

AGENDA-DRIVEN CASE MANAGEMENT

Der Fakultät für Wirtschaftswissenschaften der
Universität Duisburg-Essen vorgelegte

DISSERTATION

zur Erlangung des akademischen Grades

DOCTOR RERUM NATURALIUM

(Dr. rer. nat.)

von Marian Benner-Wickner, M. Sc.,
geboren in Detmold

Datum der mündlichen Prüfung: 14.12.2016

Erstgutachter: Prof. Dr. Volker Gruhn
Zweitgutachter: Prof. Dr. Michael Goedicke

ZUSAMMENFASSUNG

Die Dissertation beschäftigt sich mit der IT-Unterstützung wissensintensiver Geschäftsprozesse unter besonderer Berücksichtigung des Case Managements. Zu diesem Zweck vereint sie Konzepte und Methoden des Process Mining sowie des Adaptive Case Management und ergänzt sie um eigene Beiträge. Die Gesamtheit dieser Beiträge ist das Agenda-driven Case Management (adCM).

Im Gegensatz zu Routinetätigkeiten lassen sich wissensintensive Geschäftsprozesse – also Prozesse mit einem hohen Anteil an wissensintensiven Tätigkeiten, die von sogenannten Wissensarbeitern durchgeführt werden – nur schwer durch IT unterstützen. Das liegt vor allem daran, dass über den konkreten Lösungsweg und die dafür benötigten Daten nichts oder nur wenig im Vorfeld bekannt ist. Zwei wesentliche Ursachen hierfür sind, dass erstens der Ablauf von sehr vielen Parametern abhängig ist und dass zweitens diese Parameter sich auch über die Zeit verändern können. Solche Prozesse lassen sich unter anderem bei Trägern von Sozialleistungen oder in der privaten Versicherungswirtschaft beobachten. Dort steuern als Case Manager bezeichnete Wissensarbeiter komplizierte Leistungsfälle und koordinieren erforderliche Maßnahmen so, dass die Leistungen wirtschaftlich und bedarfsgerecht erbracht werden. Case Manager sind aufgrund ihrer Erfahrung, ihres breitgefächerten Fachwissens und der starken Vernetzung mit anderen Experten in der Lage, die wesentlichen Parameter der Prozesse zu erkennen, deren Veränderung stets nachzuverfolgen und den Ablauf entsprechend anzupassen.

Wie in der Dissertation gezeigt wird, können wissensintensive Prozesse nicht mit den herkömmlichen Methoden des Process Mining analysiert und mit Workflow-Managementsystemen unterstützt werden. Deshalb werden neue Konzepte und alternative Ansätze vorgestellt und erprobt, um solche Prozesse analysierbar zu machen und Case Manager bei deren Ausführung zu unterstützen. Die zentralen Beiträge der Dissertation sind ein Metamodell mit den adCM-Grundkonzepten, ein Konzept zur anwendungsübergreifenden Protokollierung der Aktivitäten eines Case Managers unter Berücksichtigung des Metamodells (Monitoring), eine Methode zur Messung von Ereignisprotokollkomplexität, eine Methode zur Erhebung von Wissen über den Prozess auf Grundlage der Ereignisprotokolle (Discovery) und eine Werkzeugarchitektur zur operativen Unterstützung von Wissensarbeitern, um das Wissen über den Prozess kontextbezogen bereitzustellen.

DANKSAGUNG

Es gibt eine Reihe von Personen, ohne die diese Dissertation nicht möglich gewesen wäre. An dieser Stelle möchte ich ihnen für ihre Unterstützung danken.

Zuerst möchte ich mich bei Volker Gruhn bedanken, der mir als Betreuer von Beginn an die wesentlichen Impulse für die thematische Ausrichtung der Arbeit gab und mich jederzeit mit hilfreichen Ratschlägen unterstützt hat. Zugleich möchte ich mich bei Matthias Book, Thomas Richter und allen anderen Mitarbeitern des Software Engineering-Lehrstuhls der Universität Duisburg-Essen bedanken. Sie haben in den zahlreichen Gesprächen und Seminaren dazu beigetragen, meinen Denkhorizont zu erweitern und ein inspirierendes Arbeitsumfeld zu schaffen. Ein besonderer Dank gilt auch Tobias Brückmann, der mir als langjähriger Mentor unermüdlich die vielleicht wertvollste Fähigkeit eines in Vollzeit beschäftigten Promotionsstudenten lehrte: die Unterscheidung zwischen „Wichtigem“ und „Dringendem“.

Das im Rahmen der Dissertation konzipierte Case Management-Werkzeug ist durch die Mitwirkung von zahlreichen Studenten umgesetzt worden. Pascal Groß, Milanko Doroski, Jens-Rene Giesen, Marvin Flock, Denis Marrese und Matthias Verholen trugen zur Wartung und Weiterentwicklung des PC-Clients bei. Yordan Terziev, Willi Koop und Stefan Nisch möchte ich für die Realisierung der Server-Komponenten danken. Ein besonderer Dank gilt Alexander Kalinowski, der als Hilfskraft sowie im Rahmen von Projekt- und Abschlussarbeiten die Entwicklung des Prototyps von der ersten Stunde an begleitet hat.

Darüber hinaus möchte ich der Bezirksverwaltung der BGN in Dortmund und insbesondere Jana Koch für die Hospitationen und die zahlreichen Expertengespräche bedanken. Auf diese Weise konnte ich detaillierte Einblicke in die Praxis des Case Managements erhalten.

Abschließend möchte ich meiner Ehefrau Mareike-Cathrine Wickner für ihre große Rücksicht und Geduld danken. Weil sie mich immer bestärkt und mir stets den nötigen Freiraum geschaffen hat, trägt sie einen wesentlichen Teil an der Verwirklichung dieser Arbeit. Mit großem Respekt und als Entschuldigung für die zahlreichen versäumten Abende und Wochenenden widme ich ihr diese Arbeit.

INHALTSVERZEICHNIS

ZUSAMMENFASSUNG.....	3
DANKSAGUNG	5
Inhaltsverzeichnis.....	6
1 Einleitung.....	14
1.1 Problemstellung.....	14
1.2 Zielstellung.....	16
1.3 Gliederung der Dissertation	16
2 Grundlagen.....	18
2.1 Geschäftsprozesse	18
2.1.1 Definition und Abgrenzung.....	18
2.1.2 Strukturierte, schwach strukturierte und unstrukturierte Prozesse.....	19
2.2 Wissensarbeiter	20
2.2.1 Definitionen der Wissensarbeit	20
2.2.2 Aktivitäten eines Wissensarbeiters.....	22
2.2.3 Wissensarbeit und schwach strukturierte Prozesse.....	24
2.3 Case Management.....	25
2.3.1 Zweck des Case Managements und zentrale Eigenschaften	25
2.3.2 Einsatzgebiete	26
2.3.3 Definitionen aus der Fachliteratur.....	27
2.3.4 Gesetzlich motivierte Definitionen	29
2.3.5 Der Case Management-Prozess.....	30
2.3.6 Begriffsabgrenzungen.....	31
3 Verwandte Forschungsansätze.....	33
3.1 Geschäftsprozessmanagement.....	33
3.1.1 Teilgebiete des Geschäftsprozessmanagements.....	34
3.1.2 Case Handling: GPM für schwach strukturierte Prozesse.....	34

3.1.3	Notationen für schwach strukturierte Prozesse	35
3.1.4	Adaptive Case Management.....	38
3.1.5	Bewertung und Bezug zur Dissertation.....	39
3.2	Process Mining.....	40
3.2.1	Überblick	40
3.2.2	Herausforderungen beim Monitoring und Discovery.....	41
3.2.3	Bewertung und Bezug zur Dissertation.....	43
3.3	Wissensbasierte Systeme.....	44
3.3.1	Case Based Reasoning.....	44
3.3.2	Case Based Planning	46
3.3.3	Bewertung und Bezug zur Dissertation.....	47
3.4	Zusammenfassung.....	48
4	Agenda-driven Case Management.....	50
4.1	Anforderungen an die Konzepte des Agenda-driven Case Management	50
4.1.1	Unterstützung des Planens	50
4.1.2	Unterstützung des Recherchierens	51
4.1.3	Unterstützung des Archivierens und Strukturierens.....	51
4.1.4	Unterstützung des Reflektierens	52
4.2	Konzeptionelle Lösungsbausteine.....	53
4.2.1	adCM-Metamodell.....	54
4.2.2	Monitoring	54
4.2.3	Discovery	55
4.3	adCM-Metamodell	56
4.3.1	Agenda	57
4.3.2	Workspace mit Artefakten	59
4.3.3	Template.....	61
4.3.4	Gesamtsicht und Fallbeispiel	63

4.4	Zusammenfassung.....	67
5	Monitoring.....	68
5.1	Problemstellung.....	69
5.1.1	Herausforderungen beim Monitoring schwach strukturierter Prozesse	69
5.1.2	Anforderungen an das adCM-Monitoring	71
5.2	Related Work.....	73
5.2.1	Monitoring im Process Mining Workflow	73
5.2.2	Monitoring zur Identifikation der aktuellen Aufgabe eines Wissensarbeiters	74
5.2.3	Ansätze zur Messung der Ereignisprotokoll-Komplexität.....	75
5.3	Lösungskonzept	76
5.3.1	Datenmodell für die anwendungsübergreifende Ereignisprotokollierung	77
5.3.2	Monitoring mit semantisch verknüpften Ereignissen	81
5.3.3	Monitoring des Fall-Kontexts.....	85
5.3.4	Monitoring-Architekturentscheidungen.....	89
5.3.5	Zusammenhang zwischen schwach strukturierten Prozessen und Ereignisprotokoll-Komplexität.....	93
5.3.6	Komplexitätsmetriken für Ereignisprotokolle.....	94
5.4	Evaluation der Protokollierung	105
5.4.1	Einführung in das Fallbeispiel.....	105
5.4.2	Evaluation der Vollständigkeit der Protokollierung	107
5.4.3	Evaluation der Verlässlichkeit der Protokollierung.....	108
5.4.4	Evaluation der Wohldefiniertheit der Protokollstruktur.....	108
5.4.5	Evaluation der Protokollierung semantisch verknüpfter Ereignisse.....	109
5.5	Evaluation der Kontextermittlung.....	110
5.5.1	Auswahl der Fallbeispiele – Szenario 1	111
5.5.2	Auswahl der Fallbeispiele – Szenario 2	111
5.5.3	Ergebnisse – Szenario 1.....	112
5.5.4	Ergebnisse – Szenario 2.....	113

5.6	Evaluation der Komplexitätsmetriken.....	114
5.6.1	Experiment mit synthetischen Log-Dateien	115
5.6.2	Ergebnisse mit Log-Dateien aus der Praxis	119
5.7	Zusammenfassung.....	123
6	Discovery	126
6.1	Problemstellung.....	127
6.1.1	Anforderungen an das Auffinden von Templates.....	127
6.1.2	Anforderungen an die Speicherung von Prozesswissen.....	128
6.2	Related Work.....	129
6.2.1	Discovery-Algorithmen im Process Mining.....	129
6.2.2	Discovery-Algorithmen variabler Geschäftsprozesse	130
6.2.3	Tree Mining: Discovery-Algorithmen für hierarchische Datenstrukturen	130
6.3	Lösungskonzept	133
6.3.1	Semantisches Discovery.....	134
6.3.2	Mehrschichtiges Datenmodell für das Discovery	137
6.3.3	Manuelle Erzeugung von Templates.....	144
6.3.4	Automatische Definition von Templates.....	145
6.4	Evaluation.....	152
6.4.1	Experimentaufbau	152
6.4.2	Ergebnisse.....	166
6.4.3	Evaluation mit einem weiteren Datensatz	172
6.5	Zusammenfassung.....	172
7	Architektur und Implementierung des adCM-Werkzeugs	175
7.1	Vorgehensweise und Begriffsklärung.....	176
7.2	Übergeordnete softwaretechnische Architektur.....	177
7.2.1	Szenarien für architekturelevante Anforderungen.....	177
7.2.2	Auswahl und Zusammensetzung der Architekturmuster	180

7.3	Design und Implementierung des Monitoring-Services.....	183
7.3.1	Szenarien für architekturelevante Anforderungen.....	184
7.3.2	Auswahl und Zusammensetzung der Architekturmuster	185
7.3.3	Implementierung der Ereignisgesteuerten Architektur	185
7.3.4	AdCM-spezifische Erweiterung des XES-Standards	186
7.3.5	Architektur und Implementierung des Monitoring-Service	187
7.4	Design und Implementierung des Discovery-Service	188
7.4.1	Szenarien für architekturelevante Anforderungen.....	188
7.4.2	Auswahl und Zusammensetzung der Architekturmuster	189
7.4.3	Implementierung der Discovery-Service	191
7.5	Design und Implementierung der Vorschlagsfunktion.....	192
7.5.1	Szenarien für architekturelevante Anforderungen.....	193
7.5.2	Auswahl und Zusammensetzung der Architekturmuster	194
7.6	Design und Implementierung von Funktionen zur Unterstützung weiterer Kernaktivitäten	196
7.6.1	Unterstützung beim Recherchieren	196
7.6.2	Unterstützung beim Archivieren.....	199
7.6.3	Unterstützung beim Strukturieren	201
7.6.4	Unterstützung beim Reflektieren.....	208
7.7	Related Work.....	213
7.7.1	Architektur des ACTIVE Knowledge Workspace	213
7.7.2	Process Mining: Process Aware Information Systems	216
7.7.3	Dyonipos-Projekt	219
7.7.4	Einordnung in eine ACM-Beispielarchitektur	221
7.8	Zusammenfassung.....	221
8	Fallbeispiel.....	223
8.1	Vorbemerkungen.....	223
8.2	Einführung in das Fallbeispiel.....	224

8.3	Beschreibung der Fallbasis.....	225
8.4	Ablauf des Beispiels.....	227
8.4.1	T _n : Ausführung des Discovery-Algorithmus	227
8.4.2	T ₀ : Arbeitsbeginn Herr Müller.....	227
8.4.3	T ₁ : Start des Falls.....	228
8.4.4	T ₂ : Moderne Therapiemethoden recherchieren	229
8.4.5	T ₃ : Therapieerfolg reflektieren	231
8.4.6	T ₄ : Psychisches Krankheitsbild	231
8.4.7	T _m : Therapie abgeschlossen	233
8.5	Zusammenfassung und Bewertung.....	235
9	Fazit und Ausblick.....	237
9.1	Fazit.....	237
9.1.1	adCM-Metamodell.....	237
9.1.2	Monitoring	238
9.1.3	Discovery	239
9.1.4	Implementierung	240
9.2	Ausblick	242
9.2.1	Durchführung verschiedener Fallstudien.....	242
9.2.2	Monitoring: Weitere Metriken für Ereignisprotokolle.....	242
9.2.3	Discovery: Weitere Analyseziele	243
9.2.4	Menschenlesbare Modellierung schwach-strukturierter Prozesse	244
9.2.5	Conformance: Definition und Einhaltung von Geschäftsregeln.....	244
10	Anhang.....	247
10.1	Sikuli-Testfall.....	248
10.2	Ereignisprotokoll	255
10.3	Semantisch verknüpftes Ereignisprotokoll (Auszug).....	266
10.4	Kontextterm-Vektoren (Auszug)	267

10.4.1	Szenario 1	267
10.4.2	Szenario 2	268
10.5	Beispiel-Template (in Graphstruktur)	269
10.6	XES-Erweiterung	271
11	Literaturverzeichnis	272
Abbildungsverzeichnis		289
Tabellenverzeichnis		292
Eidesstattliche Erklärung		294

1 EINLEITUNG

“The most important, and indeed the truly unique, contribution of management in the 20th century was the fifty-fold increase in the productivity of the MANUAL WORKER in manufacturing. The most important contribution management needs to make in the 21st century is similarly to increase the productivity of KNOWLEDGE WORK and the KNOWLEDGE WORKER.”

-- Peter Drucker

1.1 PROBLEMSTELLUNG

Die Dissertation befasst sich mit der Problemstellung, mit welchen informationstechnologischen Konzepten und Methoden Personen bei der Planung und Durchführung wissensintensiver Geschäftsprozesse unterstützt werden können. Um die spezifischen Anforderungen an die Unterstützung solcher Prozesse nachvollziehen zu können, werden die charakteristischen Eigenschaften wissensintensiver Geschäftsprozesse anhand der Metapher eines belebten Marktplatzes herausgestellt:

Das Durchqueren eines stark besuchten Marktplatzes ist eine mitunter herausfordernde Aufgabe. Zwar ist das Ziel im Vorfeld bekannt und vielleicht sogar in Sichtweite, aber der direkte Weg dorthin ist versperrt von durcheinanderlaufenden Menschen. Es ist nicht vorhersagbar, welcher Weg *genau* zum Ziel führen wird und welchen Menschen man zu welchem Zeitpunkt ausweichen muss. Hat man einen Weg gefunden und auf einer Karte präzise aufgezeichnet, so wird er sich kaum auf exakt die gleiche Weise wiederholen lassen und bildet deshalb nur den einen Fall ab. Trotzdem können verschiedene Erfahrungswerte bei zukünftigen Versuchen helfen, den Markt zu durchqueren. So kann beispielsweise eine eher grobe Skizze auf der Grundlage von vergangenen Marktbesuchen Aufschluss darüber geben, dass der Platz in der Regel rechtsseitig vom Brunnen und dann am Blumenhändler vorbei schneller passiert werden kann. Bei solchen Erfahrungswerten werden Zwischenziele oder Landmarken verwendet, um einen günstigen Weg vom Start bis zum Ziel grob zu beschreiben. Sie sind robust gegenüber Veränderungen, weil sie bei der Wegbeschreibung nicht zu sehr ins Detail gehen. Doch solche allgemeinen Erfahrungswerte, die sich auf die gesamte Strecke beziehen, helfen nicht bei unvorhergesehenen Ereignissen, die lokal isoliert irgendwo auf dem Weg auftreten können und ein Hindernis darstellen. Wenn zum Beispiel ein erfahrungsgemäß wenig besuchter Stand auf dem grob

skizzierten Weg plötzlich aufgrund eines guten Angebots von einer Menschentraube umgeben ist, sind Erfahrungswerte gefragt, die sich auf eine solch spezifische Situation beziehen. Beispielsweise kann es eine etablierte Alternative sein, diesen Bereich großzügig zu umgehen, anstatt sich durch das Gedränge zu arbeiten.

Diese Metapher beschreibt in etwa die Probleme, auf die ein sogenannter Wissensarbeiter bei der Planung und Durchführung wissensintensiver Geschäftsprozesse stößt. Im Gegensatz zu strukturierten, „fließbandähnlichen“ Prozessen, deren Ablauf in Prozessmodellen abgebildet und durch Workflow Management-Systeme unterstützt werden kann, sind bei wissensintensiven Prozessen vorab keine allgemeingültigen und vollständigen Lösungswege vorhersagbar. Deswegen erfordert die IT-Unterstützung solcher Prozesse Konzepte, die flexibler sind als ein fest vorgegebenes Prozessmodell. Analog zur Überquerung eines überfüllten Marktplatzes sind bei der Planung wissensintensiver Geschäftsprozesse grobe Informationen über mögliche „Trampelpfade“ nützlich. Dabei handelt es sich um allgemeine Erfahrungswerte, die zwar nur sehr unscharf oder abstrakt beschreibbar sind, dafür aber vom Start bis zum Ziel führen können. Auf dem Weg dorthin sind darüber hinaus auch sehr spezifische, kontextbezogene Hinweise relevant, um kurzfristig auf individuelle Ereignisse reagieren zu können. Diese Hinweise lassen sich zwar nicht auf den gesamten Prozess übertragen, erleichtern aber die Entscheidungsfindung in der konkreten Situation.

Um eine IT-Unterstützung bieten zu können, die sowohl allgemeine als auch spezifische Erfahrungswerte liefern kann, müssen zunächst Informationen über die Prozesse gesammelt und bereitgestellt werden. Für diesen Anwendungsfall bieten sich Methoden aus der Process Mining-Disziplin an. Sie beschreiben, wie Informationen über den Prozessverlauf anhand von Ereignisprotokollen von Informationssystemen gewonnen, untersucht und mithilfe von Prozessmodellen beschrieben werden können. Doch wie später noch gezeigt wird, erschweren die besonderen Eigenschaften von wissensintensiven Geschäftsprozessen die Anwendung des Process Mining. Denn aufgrund der Unvorhersehbarkeit des Lösungswegs und der benötigten Daten unterscheiden sich die einzelnen Fälle so stark voneinander, dass das daraus resultierende Modell zu komplex wird.

Um einen Wissensarbeiter bei der Planung und Steuerung eines wissensintensiven Prozesses zu unterstützen, bedarf es eines auf diese speziellen Eigenschaften zugeschnittenen Konzepts, das in Anlehnung an den Process Mining-Ansatz die vergangenen Fälle untersuchen und Erfahrungswerte daraus gewinnen kann. Weil sich, wie später gezeigt wird, die Aktivitäten eines Wissensarbeiters nicht nur auf die reine Planung zukünftiger Handlungsschritte reduzieren,

werden ebenso Funktionen zur Unterstützung weiterer Kernaktivitäten eines Wissensarbeiters benötigt.

1.2 ZIELSTELLUNG

Ziel der Dissertation ist es, Konzepte zur IT-Unterstützung von Wissensarbeitern bei der Ausübung wissensintensiver Prozesse zu erarbeiten. Dabei wird ein Schwerpunkt auf Tätigkeiten zur Planung und Steuerung von Fällen im Case Management gelegt. Die Konzepte werden unter dem Begriff „Agenda-driven Case Management“ (adCM) zusammengefasst.

1.3 GLIEDERUNG DER DISSERTATION

Die im Rahmen des Dissertationsvorhabens entwickelten Lösungsbausteine bauen auf den aktuellen Forschungsstand des Process Mining auf. Um die Bausteine zu strukturieren, wird deshalb stets auf den in (van der Aalst 2011) definierten Process Mining-Ansatz verwiesen, der in stark vereinfachter Form in der Abbildung 1 skizziert wird.

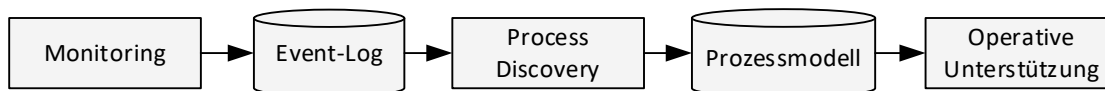


Abbildung 1: Grober Ablauf im Process Mining

Der Process Mining-Ansatz beginnt mit dem **Monitoring** der Benutzeraktionen während der Ausführung des Prozesses. Dabei werden **Ereignisdaten** wie das Öffnen eines Dokumentes oder die Neuerfassung einer Patientenakte protokolliert. Die Daten werden in **Discovery-Algorithmen** weiterverarbeitet, um **Modelle** zu generieren, die den Prozessablauf dokumentieren. Üblicherweise handelt es sich dabei um Geschäftsprozessmodelle, die dem tatsächlich ausgeführten Prozess möglichst ähnlich sind. Ausgehend von diesen Modellen können wiederum Möglichkeiten zur **operativen Unterstützung** des Akteurs entstehen, zum Beispiel das Vorschlagen des nächsten Prozessschrittes.

In Kapitel 2 werden die theoretischen Grundlagen für das Agenda-driven Case Management beleuchtet. Hier wird auch die begriffliche Einordnung der in dieser Dissertation unterstützten Geschäftsprozesse vorgenommen: Ausgehend von der Definition und Klassifikation von Geschäftsprozessen allgemein über die Definition von Wissensarbeit und den Aktivitäten eines Wissensarbeiters bis hin zur Definition des Case Managements wird die Problemdomäne immer weiter konkretisiert und eingegrenzt.

Anschließend folgt in Kapitel 3 eine Vorstellung unterschiedlicher Forschungsdisziplinen, welche die Unterstützung ebendieser Geschäftsprozesse zum Gegenstand haben. Je Disziplin werden dabei auch deren jeweiligen Stärken und Schwächen beleuchtet.

Bezugnehmend auf die zuvor genannten Schwächen oder Lücken werden in Kapitel 4 die Ziele der Dissertation konkretisiert. Anschließend werden die drei Lösungsbausteine des adCM kurz zusammengefasst und in einen funktionalen Zusammenhang gebracht. Zum Schluss werden mit dem adCM-Metamodell die wesentlichen Konzepte des Agenda-driven Case Management eingeführt, die in allen drei Lösungsbausteinen zur Erreichung der Ziele eingesetzt werden.

Je Lösungsbaustein folgt ein eigenes Kapitel. Das jeweilige Kapitel beginnt mit einer Beschreibung der adressierten Problemstellung und einer kurzen Gegenüberstellung der Related Work. Darauf folgt eine detaillierte Beschreibung der jeweiligen Lösungsansätze. Die Kapitel schließen mit einer Evaluation.

Das Kapitel 7 beschreibt die prototypische Implementierung eines Werkzeugs für den adCM-Ansatz. Das Werkzeug stellt die technische Integration der einzelnen Lösungsbausteine dar. Je Baustein werden die maßgeblichen Architektur- und Implementierungsentscheidungen vorgestellt, die bei der Realisierung des jeweiligen Konzepts getroffen wurden.

Um zu zeigen, ob die Lösungsbausteine integriert in Form des implementierten Werkzeugs einen Case Manager praktisch unterstützen können, wird in Kapitel 8 der Einsatz des Werkzeugs an einem Fallbeispiel gezeigt.

Die Dissertation wird in Kapitel 9 mit einer Zusammenfassung der gewonnenen Erkenntnisse sowie einem Ausblick auf Anknüpfungsmöglichkeiten für weiterführende Forschungsarbeiten abgeschlossen.

2 GRUNDLAGEN

In diesem Kapitel wird die Begriffswelt der Problemdomäne eingeführt. Hinführend zu den konkreten Herausforderungen bei der Unterstützung von Case Management-Prozessen wird zunächst der Geschäftsprozessbegriff beschrieben. Dabei wird auch eine Klassifikation vorgestellt, mit deren Hilfe sich die in dieser Dissertation fokussierten Prozesse einordnen lassen. Um zu zeigen, welche konkreten Herausforderungen durch die Zielstellung der Dissertation zu erwarten sind, folgt eine Annäherung an den Begriff der Wissensarbeit und die damit verbundenen Tätigkeitsfelder. Hierbei wird auch dargelegt, wie sich Wissensarbeit in die zuvor genannte Klassifikation von Geschäftsprozessen einsortieren lässt. Zum Schluss wird erläutert, was unter Case Management zu verstehen ist und welche Zusammenhänge mit der Wissensarbeit bestehen.

2.1 GESCHÄFTSPROZESSE

Gemäß der zuvor genannten allgemeinen Zielstellung zielen die in dieser Dissertation vorgestellten Beiträge auf die Unterstützung des Case Managements ab. Weil es sich dabei, wie später in Kapitel 2.3 gezeigt wird, um eine bestimmte Form von Geschäftsprozessen handelt, wird die theoretische Grundlage geschaffen, die zur Einordnung des Case Managements notwendig ist.

2.1.1 DEFINITION UND ABGRENZUNG

Dieser Arbeit wird die folgende Definition von Geschäftsprozessen nach Gadatsch zugrundegelegt:

“Ein Geschäftsprozess ist eine zielgerichtete, zeitlich logische Abfolge von Aufgaben, die arbeitsteilig von mehreren Organisationen oder Organisationseinheiten unter Nutzung von Informations- und Kommunikationstechnologien ausgeführt werden können. Er dient der Erstellung von Leistungen entsprechend den vorgegebenen, aus der Unternehmensstrategie abgeleiteten Prozesszielen. Ein Geschäftsprozess kann formal auf unterschiedlichen Detaillierungsebenen und aus mehreren Sichten beschrieben werden. Ein maximaler Detaillierungsgrad der Beschreibung ist dann erreicht, wenn die

ausgewiesenen Aufgaben je in einem Zug von einem Mitarbeiter ohne Wechsel des Arbeitsplatzes ausgeführt werden können.” (Gadatsch 2010)

Im Gegensatz zu der Definition von Scheer und Jost wird demnach unter einem Geschäftsprozess keine modellhafte Beschreibung, wie zum Beispiel ein BPMN-Diagramm (Business Process Model and Notation) oder ein Modell einer Ereignisgesteuerten Prozesskette verstanden (Scheer und Jost 1996). Gleichwohl kann, wie in der Definition von Gadatsch angedeutet, ein Geschäftsprozess mithilfe solcher Modelle beschrieben werden.

2.1.2 STRUKTURIERTE, SCHWACH STRUKTURIERTE UND UNSTRUKTURIERTE PROZESSE

Um zu verstehen, inwiefern sich die in dieser Dissertation adressierten Geschäftsprozesse von herkömmlichen Geschäftsprozessen unterscheiden, folgt eine kurze Einführung in die Klassifikation nach Loeffeler. Darin werden Geschäftsprozesse im Allgemeinen hinsichtlich der Vorhersagbarkeit der folgenden drei unterschiedlichen Aspekte eines Prozesses unterschieden: benötigte Daten, beteiligte Personen und Lösungsweg (Loeffeler et al. 1998).

Mit den benötigten Daten sind alle für die Prozessausführung relevanten Daten gemeint, die in den unterschiedlichsten Formen vorliegen können (zum Beispiel Text, Bild, Video). Unter den beteiligten Personen sind alle Akteure zu verstehen, die während des Prozesses miteinander kooperieren. Der Lösungsweg enthält alle Schritte, die im Verlauf eines Prozesses durchgeführt werden müssen, inklusive der Reihenfolge ihrer Ausführung.

Nach Loeffeler kann zu einem gegebenen Prozess entschieden werden, ob die Aspekte bereits vor der Prozessausführung bekannt sind oder nicht (plannable/unplannable). Bildet man diese Einschätzung in einem Diagramm ab, so erhält man ein dreidimensionales Koordinatensystem mit einem Würfel aus acht Sektoren (siehe Abbildung 2). Jeder dieser Sektoren repräsentiert eine Klasse der Prozessklassifikation.

Anhand dieser Klassen identifizieren Loeffeler et al. drei wesentliche Prozesstypen:

- Strukturierte Prozesse (Klasse 0)
- Ad-hoc-Prozesse (Klasse 7)
- Schwach strukturierte Prozesse (Klassen 1-6)

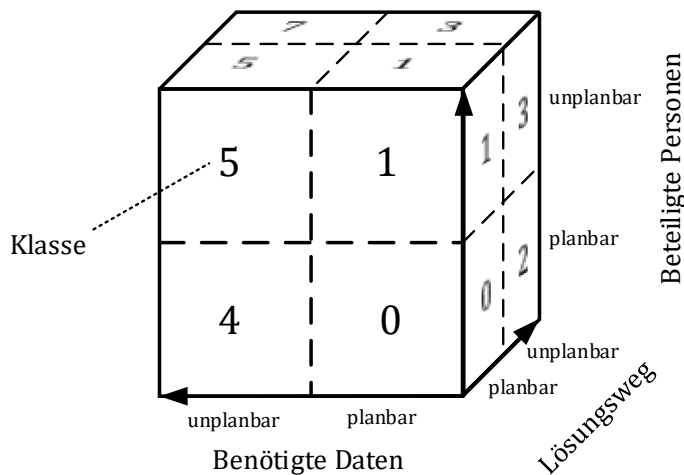


Abbildung 2: Klassifikation von Geschäftsprozessen nach (Loeffeler et al. 1998)

Die strukturierten Prozesse zeichnen sich dadurch aus, dass alle drei Aspekte vorhersagbar sind. Im Gegensatz dazu sind die ad-hoc-Prozesse so definiert, dass sich dazu kein einziger Aspekt vorhersagen lässt. Alle anderen Abstufungen hinsichtlich der Vorhersagbarkeit werden den schwach strukturierten Prozessen zugeschrieben.

2.2 WISSENSARBEITER

Wie später noch gezeigt wird, lassen sich die Aktivitäten im Case Management auf Aktivitäten abbilden, die in der Literatur der Wissensarbeit zugeordnet werden. Deshalb wird in diesem Abschnitt der Begriff „Wissensarbeit“ geklärt und wie sich Wissensarbeit in die oben beschriebene Klassifikation von Geschäftsprozessen einordnen lässt. Zur Begriffsklärung werden zunächst ausgewählte Definitionen der Wissensarbeit zusammengestellt und diskutiert mit dem Ziel, typische Tätigkeiten eines Wissensarbeiters zusammenzustellen. Dies erfolgt vorbereitend auf die Konkretisierungen der Tätigkeiten bezogen auf das Case Management in Kapitel 2.3.

2.2.1 DEFINITIONEN DER WISSENSARBEIT

Der Begriff des Wissensarbeiters wurde erstmals in einer 1969 erstausgegebenen Publikation von Peter Drucker erwähnt. Darin definiert er den Wissensarbeiter wie folgt:

„An employee whose major contribution depends on employing his knowledge rather than his muscle power and coordination, frequently contrasted with production workers who employ muscle power and coordination to operate machines.“ (Drucker 1992, S. 564)

Diese Definition wird in der Literatur zwar oft herangezogen, enthält aber noch keine konkreten Aktivitäten eines Wissensarbeiters und ist deshalb für eine Ableitung von

Unterstützungsbedarfen eher untauglich. Durch spätere empirische Studien konnten vier basale Aktivitäten eines Wissensarbeiters herausgestellt werden (Heisig 1999): Wissen **erzeugen**, **speichern**, **verteilen** und **anwenden**. Drei dieser Aktivitäten lassen sich auch in der Definition von Wissensarbeitern nach Swenson wiederfinden:

„People who have a high degree of expertise, education, or experience and the primary purpose of their job involves the creation, distribution, or application of knowledge“ (Swenson 2010a)

Die Feststellung solcher Aktivitäten der Wissensarbeit ermöglicht bereits eine gewisse Konkretisierung der typischen „Use Cases“ eines Wissensarbeiters. Doch noch immer bleibt unklar, auf welche Weise Wissen erzeugt wird und welche untergeordneten Aktivitäten hierbei zum Tragen kommen. Zur Konkretisierung dieser Aufgaben werden im Folgenden weitere Publikationen herangezogen.

Nach (Willke et al. 2001) versteht man unter Wissensarbeit ein „reflexivstrategisches **Bewerten**, **Entscheiden** und **Handeln**, für das sowohl Informationen als auch die Kenntnis und Einordnung von Sachverhalten [...] wichtig ist.“ Hier wird nun erstmals auch deutlich, dass neben dem reinen Erzeugen von Wissen auch dessen Evaluation wichtig ist. Zudem wird die Notwendigkeit klar, dass zuvor Informationen gesammelt und einsortiert werden müssen. Zu einer detaillierteren Aufschlüsselung von Tätigkeiten in Zusammenhang mit der Informationsverarbeitung trägt Luczak bei, indem er unter anderem zwischen dem **Aufnehmen**, **Kombinieren** und **Verknüpfen** von Informationen unterscheidet (Luczak 1998). Eine sehr komplexe und differenzierte Beschreibung der Aktivitäten eines Wissensarbeiters wird von North und Güldenbergs aufgestellt. Sie identifizieren fünf sogenannte „Komponenten“ der wissensbasierten Wertschöpfung, die während des Prozesses wechselwirkend und ohne explizite Reihenfolge zur Zielerreichung vom Wissensarbeiter eingesetzt werden (North und Güldenbergs 2008).

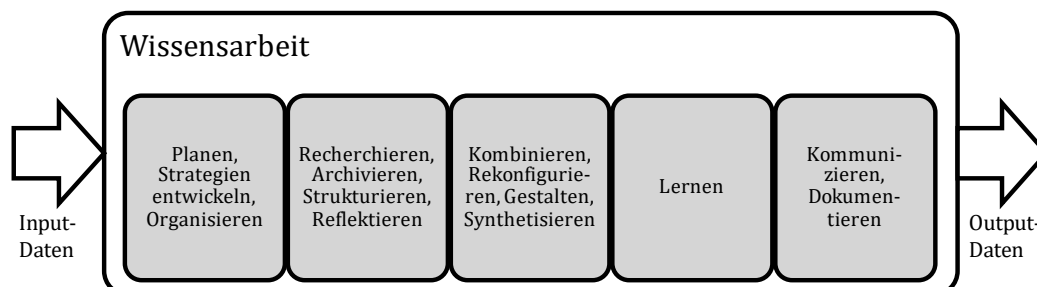


Abbildung 3: Tätigkeitsfelder während der Wissensarbeit nach (North und Güldenbergs 2008)

Weil es sich bei diesem Ansatz (siehe Abbildung 3) um eine klar abgegrenzte und detaillierte Aufstellung von Aktivitäten handelt, wird er in dieser Arbeit als Bezugspunkt zur Beschreibung der Kernaktivitäten eines Wissensarbeiters angeführt.

2.2.2 AKTIVITÄTEN EINES WISSENSARBEITERS

Im weiteren Verlauf wird eine Auswahl der in (North und Güldenbergs 2008) aufgestellten Aktivitäten detaillierter beschrieben. Dabei wird auch jeweils kurz darauf eingegangen, welche Herausforderungen bei der IT-Unterstützung dieser Aktivitäten entstehen. Die Auswahl erfolgt beziehungsweise auf die in Kapitel 4.1 festgelegte Fokussierung der Dissertation. Weil die drei Komponenten „Lernen“, „Kommunizieren“ und „Synthetisieren“ darin nicht enthalten sind, werden sie im Folgenden ausgeblendet werden.

2.2.2.1 Planen

Eine der fünf Komponenten der Wissensarbeit ist nach North und Güldenbergs das Planen, Organisieren und Entwickeln von Strategien (siehe Abbildung 3). Durch diese Aktivitäten „erfolgt die gedankliche Vorwegnahme zukünftigen Handelns. Mentale Modelle entstehen, die beim Handeln dann ständig fortgeschrieben werden“ (North und Güldenbergs 2008). Diese Definition von Planungsaktivitäten weist auf zwei Punkte hin: Erstens handelt es sich um eine gedankliche Aktivität, deren Ergebnis in einem mentalen Modell festgehalten und nicht notwendigerweise externalisiert (Nonaka und Takeuchi 1997) wird. Zweitens sind diese Modelle nicht festgeschrieben, sondern Gegenstand einer kontinuierlichen Weiterentwicklung. In beiden Punkten unterscheidet sich Wissensarbeit deutlich von Routinetätigkeiten. Deren Arbeitsschritte können im Gegensatz zur Wissensarbeit schon im Vorfeld geplant und dadurch auch festgeschrieben werden, beispielsweise in Form von Prozessmodellen.

Nach dieser Definition sollte eine IT-Unterstützung von Planungsaktivitäten die folgenden Punkte beachten:

1. Eine auf einen Plan angewendete Analyse muss berücksichtigen, dass der Plan nicht notwendigerweise vollständig ist. Dabei spielt es keine Rolle, ob der Wissensarbeiter nicht alle zukünftigen Aktivitäten externalisiert hat, oder ob sie zu dem Analysezeitpunkt noch unbekannt sind.
2. Eine Datenstruktur zur Externalisierung des Plans muss berücksichtigen, dass der Plan sich ständig verändern kann. Sie muss also entsprechend flexibel sein.

2.2.2.2 Recherchieren

Die Aktivität „Recherchieren“ umfasst Tätigkeiten zur Suche nach benötigten Informationen. Nach (Loeffeler et al. 1998) sind die dafür benötigten Datenquellen nicht unbedingt im Vorfeld bekannt: Im Gegensatz zu einem Sachbearbeiter mit klar definierten Aufgabenbereichen, der in einem strukturierten Prozess mithilfe betrieblicher Informationssysteme alle benötigten Informationen im Zugriff hat, ändert sich der Informationsbedarf eines Wissensarbeiters während der Lösung seiner aktuellen Aufgabenstellung kontinuierlich. Folglich arbeitet ein Wissensarbeiter während einer Recherche mit einer Vielzahl verschiedener Typen von Informationsquellen. Das stellt eine Herausforderung dar, weil viele dieser Typen eine eigene Benutzerschnittstelle besitzen, die Daten in einer unterschiedlichen Struktur zur Verfügung stellen und nicht im Unternehmen selbst, sondern extern bereitgestellt werden – mit den entsprechenden Konsequenzen für die Zugriffsrechte.

Um diese Zusammenhänge zu verdeutlichen, wird ein Beispiel-Szenario eingeführt. Darin füllt ein Hausarzt die Rolle des Wissensarbeiters aus. Diese Berufsgruppe wird nach Wendt den sogenannten „Primary Care Providers“ zugeordnet (Wendt 2008a, S. 227). Ihre Aufgabe ist es, im Sinne eines zentralen Ansprechpartners für den Patienten dessen Behandlungsbedarf zu prüfen und alle weiteren medizinischen Maßnahmen zu veranlassen. Hierfür sind alle Informationen über die aktuelle Diagnose oder die Krankheitsgeschichte potentiell wertvoll. Es spielt in diesem Zusammenhang für den Arzt keine Rolle, über welche Quelle er diese Informationen bekommt – abgesehen davon, dass sie gesetzlich zulässig ist. Um im weiteren Therapieverlauf entsprechend der Diagnose passende Fachärzte zu finden, konsultiert er unter anderem eine Webseite der zuständigen Ärztekammer. Zusätzlich öffnet er eine in der Gemeinschaftspraxis gepflegte Tabelle mit bewerteten Spezialisten und Krankenhäusern aus der Region, um Erfahrungswerte über die Fachärzte einzuholen und weitere lokal ansässige Ärzte zu finden. Alle genannten Datenquellen sind jeweils über unterschiedliche Informationssysteme oder Anwendungen erreichbar, haben eine unterschiedliche Struktur und sind teils im Unternehmen selbst, teils extern bereitgestellt.

2.2.2.3 Archivieren und Strukturieren

Nach (North und Güldenberger 2008) ist es eine zentrale und wiederkehrende Aufgabe von Wissensarbeitern, Rechercheergebnisse in das vorhandene Wissen beziehungsweise die vorhandenen Informationen einzuordnen. Hierzu müssen die Informationen abgelegt und in eine Struktur gebracht werden können. Diese Aktivität ist nicht zu verwechseln mit dem Dokumentieren, denn das Dokumentieren erfolgt in der Regel adressatenbezogen und ist deshalb zusammen mit der Tätigkeit „Kommunizieren“ einer eigenen Komponente zugeordnet (siehe Abbildung 3).

Wie schon beim Recherchieren, so ist auch bei der strukturierten Ablage die Vielfalt an unterschiedlichen Informationsquellen eine Herausforderung, denn die Informationen liegen je nach Quelle in unterschiedlichen Formaten vor. Der Hausarzt aus dem vorangegangenen Beispiel kann zwar Webseiten wechselseitig mittels Hyperlinks in Bezug setzen. Webseiten sind aber nicht dafür vorgesehen, als einzelne Dateien in einer digitalen Akte abgelegt zu werden. Umgekehrt ist es zwar keine technische Herausforderung, PDF-Dokumente abzulegen, aber das Verknüpfen solcher Dokumente ist technisch kompliziert.

2.2.2.4 Reflektieren

Eine weitere Tätigkeit des Wissensarbeiters ist, die recherchierten und ggfs. bereits in einer Ablage strukturierten Informationen zu reflektieren (North und Güldenbergh 2008). Hierzu werden verschiedene Unterstützungsfunktionen seitens des Betriebssystems oder der Office-Anwendungen verwendet. Dazu zählen Möglichkeiten zum Hervorheben, Verschlagworten, Kommentieren oder Bewerten von Informationen. Die auf diese Weise hinzugefügten Metadaten werden im weiteren Verlauf der Dissertation als „Annotationen“ bezeichnet. Sie können auf zwei unterschiedlichen Ebenen hinzugefügt werden: auf *Dateiebene*, wo sie sich auf das Dokument, die E-Mail oder die Webseite als Ganzes beziehen, und auf *Inhaltsebene*, wo sie sich auf bestimmte Textpassagen, Bilder oder andere Inhaltsformen beziehen (Benner-Wickner et al. 2013).

Auch in diesem Zusammenhang lassen sich durch die Vielfalt an unterschiedlich strukturierten Informationsquellen Herausforderungen ableiten. So bieten zwar viele Anwendungen zur Darstellung von Dokumenten sehr ähnliche Funktionen zum Kommentieren und Hervorheben von Textpassagen, aber in Anwendungen zur Darstellung von Webseiten fehlen solche Funktionen. Die Ursache hierfür liegt darin, dass Metadaten üblicherweise in den Dokumenten selbst gespeichert werden. Ein Wissensarbeiter wie der oben erwähnte Hausarzt hat aber auf extern bereitgestellten Webseiten keinen Schreibzugriff. Ein ähnliches Problem besteht beim Annotieren auf Dateiebene. Hier bieten moderne Betriebssysteme zwar Funktionen zur Bewertung an, jedoch nur bezogen auf bestimmte Medienformate wie Bilder oder Videos (Benner-Wickner et al. 2013).

Die in diesem Abschnitt beschriebenen Aktivitäten werden im weiteren Verlauf auch als „wissensintensive Tätigkeiten“ zusammengefasst.

2.2.3 WISSENSARBEIT UND SCHWACH STRUKTURIERTE PROZESSE

Nach Swenson (2010a) gibt es einen starken Zusammenhang zwischen Wissensarbeit und schwach strukturierten Prozessen. Demnach geht Wissensarbeit stets mit schwach strukturierten

Prozessen einher, weil der Verlauf von Fall zu Fall unterschiedlich ist. Eine weitere Erklärung für diesen Zusammenhang bietet die Definition eines Wissensarbeiters nach Kelley. Er beschreibt die Tätigkeiten eines Wissensarbeiters als herausfordernd und nicht wiederholbar:

„Many gold-collar workers [Wissensarbeiter – Anm. d. Verf.] don’t know what they will do next, when they will do it, or sometimes even where“ (Kelley 1985).

Auch Swenson geht bei der Beschreibung der Eigenschaften von Wissensarbeit auf den Aspekt der Unvorhersehbarkeit und auf eine fehlende Reproduzierbarkeit ein (Swenson 2010a). Diese Definitionen decken sich mit den Eigenschaften schwach strukturierter Prozesse nach (Loeffeler et al. 1998). Demnach sind Prozesse mit einem hohen Anteil an Wissensarbeit definitionsgemäß schwach strukturiert.

2.3 CASE MANAGEMENT

Entsprechend der in Kapitel 1.2 aufgeführten Zielstellung wird in der Dissertation ein Schwerpunkt auf Konzepte zur Unterstützung des Case Managements gelegt – einem Ansatz, der aus den USA und Großbritannien stammt und im deutschsprachigen Raum Ende der 1980er Jahre von Wolf Reiner Wendt eingeführt wurde (Löcherbach 2009). In diesem Kapitel wird zunächst beschrieben, welche Probleme das Case Management löst und in welchen Domänen es eingesetzt wird. Es folgen Definitionen des Case Managements aus der Fachliteratur und aus der deutschen Gesetzgebung. Einen tieferen Einblick in die Arbeitsweise eines Case Managers liefert der Case Management-Prozess. Abschließend wird gezeigt, wie das Case Management in die Begriffe „Geschäftsprozess“ und „Wissensarbeit“ eingeordnet wird.

2.3.1 ZWECK DES CASE MANAGEMENTS UND ZENTRALE EIGENSCHAFTEN

Das Case Management ist ein Konzept für Kostenträger zum Zweck einer möglichst optimalen, auftragsgemäßen Erbringung von personenbezogenen Leistungen im Einzelfall (Wendt 2008b, S. 17). Es kommt unter anderem dann zur Anwendung, wenn die Steuerung eines einzelnen Leistungsfalls einer „komplexen Problematik mit einer Mehrzahl von Beteiligten“ unterliegt (ebd.). Case Management wird demnach *nicht* für die Steuerung einzelner Leistungen benötigt, beispielsweise im Rahmen eines regulären Arztbesuchs, der in keinem übergeordneten therapeutischen Zusammenhang steht.

Unabhängig von den teils domänenspezifischen Ausprägungen des Ansatzes, die im weiteren Verlauf dieses Kapitels beschrieben werden, teilen Case Management-Prozesse die folgenden

Gemeinsamkeiten: Das detaillierte Vorgehen ist **keine Routinetätigkeit**, sondern wird individuell auf die Situation der leistungsberechtigten Person, dem Klienten, zugeschnitten. Es wird für jeden Klienten, der je nach Domäne unterschiedlich definiert ist, ein **zentraler Ansprechpartner** zugeteilt. Der Ansprechpartner ist – im Gegensatz zu Sachbearbeitern, die nur für ausgewählte Teilbereiche eines Prozesses verantwortlich sind – für die Steuerung des gesamten Prozessverlaufs verantwortlich. Es handelt sich bei ihnen also um Mitarbeiter mit einem breit aufgestellten fachlichen Wissen. Um bei wichtigen Entscheidungen dennoch auf eine tiefergehende Fachlichkeit zurückgreifen zu können, sind sie mit Experten wie spezialisierten Gutachtern oder Ärzten **stark vernetzt** und erschließen sich das erforderliche Wissen. Sie arbeiten **wissensbasiert** und nutzen **formelle und informelle Ressourcen** (Wendt 2008b, S. 154). Alle Informationen werden dabei in einer Fallakte dokumentiert. Mit dieser Methode soll die nötige Flexibilität und Übersicht über den Prozessverlauf sichergestellt werden. Diese zentralen Ansprechpartner werden mitunter domänen- und organisationsspezifisch unterschiedlich benannt (beispielsweise „Reha-Manager“ oder „Berufshelfer“). Sie werden im Folgenden einheitlich unter der übergeordneten Bezeichnung Case Manager zusammengefasst.

2.3.2 EINSATZGEBIETE

Case Management ist eine domänenübergreifende Management-Methode und kann überall dort Anwendung finden, wo der oben beschriebene Zweck nutzbringend ist. Nach Wendt trifft das vor allem auf alle Bereiche personenbezogener Dienstleistungen zu, in denen das Verfahren auf die Situation des Nutzers interdisziplinär abgestimmt und gesteuert werden muss (Wendt 2008b, S. 153). Die Tabelle 1 zeigt einen Überblick über den Einsatz von Case Management in unterschiedlichen Domänen.

Tabelle 1: Einsatzgebiete des Case Managements nach (Swenson 2010b; Wendt 2008b; Mühlum und Gödecker-Geenen 2003; McCauley 2010)

Domänen	Beispielprozesse	Motivation / Nutzen	Klienten
Sozial- und Gesundheitsdienste	Reha-Management (SGB IX); Arbeitsvermittlung (SGB II); Kinder- und Jugendhilfe (SGB VIII); Altenpflege; Psychiatrie; Integrierte medizinische Versorgung	Wirksamkeit der Unterstützungsleistungen; Beschleunigung der beruflichen und sozialen Integration; Prävention vor hohen Folgekosten (Pflege- beziehungsweise Sozialleistungen)	Patient; Arbeits-suchender

Versicherungswirtschaft	Personenschadenmanagement; Underwriting	Klärung des Leistungsfalls; Zuordnung von Kosten und Nutzen der Leistungen	versicherte Person
Justiz; Strafverfolgung	Ermittlungsverfahren; Straffälligen- und Bewährungshilfe	Motivfeststellung; Aufklärung von Straftaten; Resozialisierung	angeklagte Person
Kundendienst; Servicedesk	Incident-, Change- und Problem-Management	Verifizieren und Koordinieren von Service-Anfragen durch einen zentralen Ansprechpartner (Single Point of Contract)	Kunde

Die Tabelle macht deutlich, dass in den Sozial- und Gesundheitsdiensten die gezielte Steuerung einzelner Fälle durch einen zentralen, vernetzten Ansprechpartner maßgeblich durch die Gesetzgebung motiviert ist. Case Management soll hier die Wirksamkeit der Unterstützungsleistungen sicherstellen. In der Versicherungswirtschaft ist die Anwendung von Case Management weniger gesetzlich, sondern vielmehr aus einer betriebswirtschaftlichen Perspektive motiviert. Hier soll begutachtet werden, ob es sich tatsächlich um Leistungsfälle handelt und ob die Kosten auch stets einem entsprechenden Nutzen gegenübergestellt werden. Ähnlich verhält es sich in der Domäne der Justiz, wo die von Wiederholungstätern verursachten Kosten im Strafvollzug durch möglichst effektive Resozialisierungsprogramme reduziert werden sollen. Eine etwas abgewandelte Form des Case Managements findet in der Strafverfolgung statt. Hier geht es zwar nicht um die Steuerung von Leistungen. Dennoch sind in dieser Branche zur Aufklärung von Straftaten sehr ähnliche Analysen des sozialen Umfelds der betroffenen Person nötig wie in den Sozial- und Gesundheitsdiensten. Neben diesen Branchen wird Case Management auch im Kundendienst bei der Klärung und Bearbeitung von Serviceanfragen eingesetzt. Hier stellt der Case Manager gegenüber dem Kunden den „Single Point of Contract“ dar und koordiniert zentral die zu erledigenden Dienstleistungen, um den Fall zu klären und eine Lösung anzubieten. Das Case Management soll in diesem Zusammenhang unter anderem verhindern, dass der Kunde während der Lösungsfindung dasselbe Problem vielen unterschiedlichen Sachbearbeitern erörtern muss.

2.3.3 DEFINITIONEN AUS DER FACHLITERATUR

Dermot McCauley, Author mit Berufserfahrung und verschiedenen Publikationen in diesem Themengebiet, definiert Case Management wie folgt:

„Case Management is the management of long-lived collaborative processes that require coordination of knowledge, content, correspondence, and resources to achieve an objective or goal. The path of execution cannot be predefined. Human judgment is required in determining how to proceed, and the state of a case can be affected by external events.“ (McCauley 2010)

Anhand dieser Definition lässt sich Case Management unmittelbar sowohl mit der Wissensarbeit als auch mit schwach strukturierten Prozessen in Zusammenhang bringen. Denn zum einen nennt McCauley darin typische Aktivitäten der Wissensarbeit, zum anderen weist er darauf hin, dass die Prozessschritte nicht im Vorfeld festgelegt werden können.

Nach Swenson, einem Autor und Herausgeber zahlreicher Publikationen zur IT-Unterstützung von Case Management, ist der Begriff wie folgt definiert:

“[Case Management is] a method or practice of coordinating work by organizing all of the relevant information into one place—called a case. The case becomes the focal point for assessing the situation, initiating activities and processes, as well as keeping a history record of what has transpired.“ (Swenson 2010a)

Eine genauere Betrachtung dieser Definition ermöglicht eine Gegenüberstellung der darin beschriebenen Aktivitäten mit den fünf Komponenten der Wissensarbeit nach (North und Güldenbergh 2008): Das Koordinieren von Tätigkeiten ist eine Aktivität, die im engeren Sinne dem **Planen** und im entfernteren Sinne auch dem **Kommunizieren** zugeordnet werden kann. Das Organisieren relevanter Informationen an einem zentralen Ort entspricht zum einen dem **Recherchieren**, denn um relevante Informationen zu erhalten, müssen diese zunächst zusammengetragen werden. Es entspricht zum anderen dem **Strukturieren**, denn die Informationen müssen in die Fallakte einsortiert werden. Mithilfe der Fallakte wird die Situation bewertet, was dem **Reflektieren** zugeordnet werden kann. Zuletzt wird das Geschehene aufgezeichnet, was dem **Archivieren** und **Dokumentieren** entspricht.

Diese Gegenüberstellung macht deutlich, dass das Case Management eine spezielle Form der Wissensarbeit ist. Die IT-Unterstützung von Case Management erfordert demnach nicht die Unterstützung aller Komponenten der Wissensarbeit nach (North und Güldenbergh 2008). Das „Lernen“ ist beispielsweise keine primäre Aufgabe eines Case Managers. Außerdem wird deutlich, dass die zentrale Organisation der Daten in einer Fallakte eine wesentliche Rolle bei der Koordination der Arbeitsschritte spielt. Die Bereitstellung einer solchen Datenstruktur muss demnach bei der IT-Unterstützung von Case Management berücksichtigt werden.

In der Definition von Swenson heißt es weiter:

“Beyond this generic definition, case management has specific meanings in the medical care, legal, and social services fields.” (Swenson 2010a)

In Deutschland sind domänenspezifische Definitionen von Case Management zumeist aus der Gesetzgebung heraus entstanden. Das nachfolgende Kapitel geht beispielhaft auf zwei Bereiche ein: die Arbeitsvermittlung und die gesetzliche Unfallversicherung.

2.3.4 GESETZLICH MOTIVIERTE DEFINITIONEN

Leistungsträgern gemäß § 12 SGB I („Erstes Buch Sozialgesetzbuch“) dient Case Management der praktischen Umsetzung gesetzlicher Verpflichtungen. Je nach Domäne, etwa dem Gesundheitswesen oder der gesetzlichen Rentenversicherung, werden diese Verpflichtungen in unterschiedlichen Sozialgesetzbüchern definiert. In der Domäne der Arbeitsvermittlung beispielsweise wurde Case Management im Rahmen der Novellierung der Sozialgesetzbücher durch das „Vierte Gesetz für moderne Dienstleistungen am Arbeitsmarkt“ (umgangssprachlich bekannt als „Hartz IV-Gesetz“) gesetzlich festgeschrieben. In der Gesetzesbegründung heißt es:

„Im Rahmen des Fallmanagements wird die konkrete Bedarfslage des Betroffenen erhoben; darauf aufbauend wird dann ein individuelles Angebot unter aktiver Mitarbeit des Hilfebedürftigen geplant und gesteuert. Dabei spielt der Grundsatz „Fördern und Fordern“ eine zentrale Rolle“ (BT-Drs 16/1516 2003, S. 44).

Im zweiten Sozialgesetzbuch (SGB II, § 14) wird hierzu ein persönlicher Ansprechpartner definiert.

Ein weiteres Beispiel für die gesetzlich motivierte Anwendung von Case Management findet sich in der Domäne der gesetzlichen Unfallversicherung, wo gemäß dem dritten Kapitel des SGB VII umfassende Leistungen zur Heilbehandlung und Teilhabe am Arbeitsleben und am Leben in der Gemeinschaft zu steuern sind. Eine solche Steuerung wird in den gesetzlichen Unfallversicherungen von sogenannten Berufshelfern (vgl. auch SGB IX § 111) und Reha-Managern übernommen. Dabei müssen gemäß SGB IX § 10 auch Leistungen unterschiedlicher Träger koordiniert werden.

„[...] der nach § 14 leistende Rehabilitationsträger [ist] dafür verantwortlich, dass die beteiligten Rehabilitationsträger im Benehmen miteinander und in Abstimmung mit den Leistungsberechtigten die nach dem individuellen Bedarf voraussichtlich erforderlichen Leistungen funktionsbezogen

feststellen und schriftlich so zusammenstellen, dass sie nahtlos ineinandergreifen. Die Leistungen werden entsprechend dem Verlauf der Rehabilitation angepasst und darauf ausgerichtet, den Leistungsberechtigten unter Berücksichtigung der Besonderheiten des Einzelfalls die [...] umfassende Teilhabe am Leben in der Gesellschaft zügig, wirksam, wirtschaftlich und auf Dauer zu ermöglichen. Dabei sichern die Rehabilitationsträger durchgehend das Verfahren entsprechend dem jeweiligen Bedarf und gewährleisten, dass die wirksame und wirtschaftliche Ausführung der Leistungen nach gleichen Maßstäben und Grundsätzen erfolgt.“

Zur Umsetzung dieser Aufgaben hat der zuständige Dachverband (DGUV) einen Handlungsleitfaden mit den Trägern abgestimmt. Demnach ist das Case Management im Rehabilitationsmanagement entlang der Gesetzeslage definiert als

„[...] die umfassende Planung, Koordinierung und zielgerichtete, aktivierende Begleitung der medizinischen Rehabilitation und aller Leistungen zur Teilhabe am Arbeitsleben und am Leben in der Gemeinschaft auf der Grundlage eines individuellen Reha-Plans unter partnerschaftlicher Einbindung aller am Verfahren Beteiligten“ (DGUV 2014).

Alle in diesem Abschnitt genannten Definitionen heben die planerischen Tätigkeiten im Case Management hervor und unterscheiden sich im Wesentlichen nur durch die Definition der gesteuerten Leistungen. Sie unterstreichen die Bedeutsamkeit, Case Manager bei der Ausführung von planerischen Tätigkeiten durch IT zu unterstützen.

2.3.5 DER CASE MANAGEMENT-PROZESS

Zur Konkretisierung des Case Managements beschreibt Wendt einen Prozess aus fünf Phasen, die im Folgenden einzeln beschrieben werden. Um den Bezug zur Wissensarbeit aufrechtzuerhalten, werden die Phasen dabei jeweils den Aktivitäten der Wissensarbeit zugeordnet.



Abbildung 4: Die fünf Phasen des Case Management-Prozesses nach (Wendt 2012)

Beim Outreach handelt es sich um die Eingangsphase in die Fallsteuerung (Wendt 2012). Hier findet die Eingangsprüfung der zu Prozessbeginn vorhandenen Daten statt. In der Versicherungswirtschaft bedeutet das die erste Sichtung der Falldaten sowie eine vorläufige Klärung, ob es sich um einen zu steuernden Versicherungsfall handelt. Im Gesundheitswesen geht es hingegen um die vorläufige Erfassung der subjektiven Hilfsbedürftigkeit. Bezugnehmend auf

die Komponenten der Wissensarbeit von (North und Güldenbergl 2008) werden hier also die Eingangsdaten **strukturiert abgelegt** und **reflektiert** (vgl. Abbildung 3).

Das anschließende Assessment fasst Wendt zusammen als „eingehende Klärung der Problemlage und Bedarfsfeststellung“ (Wendt 2012, S. 510). Hierzu werden die zuvor gesammelten Eingangsdaten also um tiefergehende Feststellungen ergänzt wie zum Beispiel die Lebensumstände des Klienten. Nach (North und Güldenbergl 2008) finden hier hauptsächlich **Recherchetätigkeiten** statt, deren Ergebnisse **abgelegt** und mit den vorliegenden Daten **in Bezug gesetzt** werden (vgl. Abbildung 3).

Es folgt eine Planungsphase, in der nach Wendt mit zuvor identifizierten Experten sowie dem Klienten selbst Ziele vereinbart werden. Dabei wird auch festgehalten, wie die Beteiligten auf dem Weg zur Erreichung dieser Ziele eingebunden werden. Dabei handelt es sich um einen Prozess, der Korrekturen oder neue Vereinbarungen hervorbringen kann (Wendt 2012). Vergleicht man diese Beschreibung mit den Aktivitäten der Wissensarbeit nach (North und Güldenbergl 2008), so lässt sich ableiten, dass ein Case Manager in dieser Phase schwerpunktmäßig mit **Planungstätigkeiten** konfrontiert ist (vgl. Abbildung 3).

Unter der Implementierung ist die Durchführung der geplanten Maßnahmen zu verstehen. In dieser Phase wird vom Case Manager verlangt, dass er die Umsetzung des Plans überwacht (Monitoring). In diesem Zusammenhang übernimmt er auch das Qualitäts- und Beschwerdemanagement (Wendt 2012). Verglichen mit den anderen Phasen ist der Case Manager hier in einer eher passiven Rolle, in der er in regelmäßigen Abständen den Planungsstand verfolgt und nur bei Bedarf tätig wird. Es lassen sich demnach keine Schwerpunkte bezogen auf die Tätigkeiten in der Wissensarbeit feststellen.

Mit der Evaluationsphase werden die Fortschritte durch die Beteiligten eingeschätzt. Je nach Ergebnis wird der Fall abgeschlossen oder es findet eine neue Bedarfsfeststellung in einem Re-Assessment statt (Wendt 2012). In dieser Phase werden Informationen in der Fallakte, darunter vor allem die Zielvereinbarungen, **reflektiert** (vgl. Abbildung 3).

2.3.6 BEGRIFFSABGRENZUNGEN

Zur Klärung der weiteren Verwendung der Begriffe Wissensarbeiter, Case Manager, Fall und Geschäftsprozess folgen zwei Begriffsabgrenzungen.

Begriffsabgrenzung Wissensarbeiter / Case Manager

Wenn im weiteren Verlauf dieser Arbeit von einem **Wissensarbeiter** die Rede ist, dann handelt es sich um eine Person, deren Kern-Betätigungsfeld sich in die unter 2.2.3 definierten Aktivitäten einordnen lässt. Beispiele für solche Wissensarbeiter sind Wissenschaftler, Berater oder Journalisten.

Wenn von einem **Case Manager** die Rede ist, dann handelt es sich um eine besondere Ausprägung eines Wissensarbeiters, dessen Kern-Betätigungsfeld sich auf die Steuerung von Fällen in der Leistungserbringung konzentriert. Zu diesem Zweck wendet er den oben beschriebenen Case Management-Prozess an. Beispiele für Case Manager sind Mitarbeiter im Reha-Management der gesetzlichen Unfallversicherer oder Mitarbeiter von Kinder- und Jugendhilfeeinrichtungen.

Begriffsabgrenzung Prozessinstanz / Geschäftsprozess / Case (Fall)

In Abgrenzung zum Begriff Geschäftsprozess aus Kapitel 2.1 wird im weiteren Verlauf der Arbeit die folgende Definition für den Begriff **Fall** nach Swenson zugrundegelegt:

“The name given to the specific situation, set of circumstances, or initiative that requires a set of actions to achieve an acceptable outcome or objective. Each case has a subject that is the focus of the actions—such as a person, a lawsuit, or an insurance claim—and is driven by the evolving circumstances of the subject.” (Swenson 2010a)

Anhand dieser Definition wird ein wichtiger Unterschied zwischen einem Fall und einem Geschäftsprozess deutlich: Es handelt sich bei einem **Fall** um eine konkrete, spezifische Situation. Es handelt sich nicht um eine Abstraktion von wiederholbaren Aktivitäten in der Wertschöpfung. Insofern ist ein Fall eher zu vergleichen mit einer konkreten Ausprägung oder individuellen **Instanz** eines **Geschäftsprozesses**, beispielsweise der Erfassung eines Risikolebensversicherungsvertrags mit Barrente für eine bestimmte zu versichernde Person. Beim diesem Vergleich lassen sich zwar Gemeinsamkeiten feststellen wie zum Beispiel das Verfolgen eines abgestimmten Ziels. Bei der Bearbeitung eines Falls aber ist zum einen das Ziel nicht unbedingt über die betriebswirtschaftliche Strategie definiert, und zum anderen dreht es sich in einem Fall immer um ein zentrales Subjekt. Zudem ist für einen Fall keine zeitlich logische Abfolge definiert und es ist nicht gefordert, dass er sich arbeitsteilig durchführen lässt.

3 VERWANDTE FORSCHUNGSANSÄTZE

3.1 GESCHÄFTSPROZESSMANAGEMENT

In diesem Kapitel wird darauf eingegangen, was unter Geschäftsprozessmanagement (GPM) zu verstehen ist und was diese Disziplin dazu beiträgt, die zuvor klassifizierten Ausprägungen von Geschäftsprozessen zu unterstützen. Für einen Überblick über den gesamten GPM-Forschungsstand, der aufgrund seines Umfangs an dieser Stelle nicht geleistet werden kann, wird auf die einschlägige Literatur verwiesen, wie zum Beispiel (Rosing et al. 2014; Dumas 2013). Stattdessen werden gezielt nur die Teilgebiete und Beiträge hervorgehoben, die sich auf die Zielstellung der Dissertation beziehen.

Swenson und von Rosing haben in (Swenson und von Rosing 2014) unterschiedliche Definitionen von Geschäftsprozessmanagement in einer gemeinsamen Definition zusammengefasst:

„Business Process Management (BPM) is a discipline involving any combination of modeling, automation, execution, control, measurement and optimization of business activity flows, in support of enterprise goals, spanning systems, employees, customers and partners within and beyond the enterprise boundaries.“

In diese weitgefasste Definition sind Methoden zur IT-gestützten Ausführung von Geschäftsprozessen eingeschlossen. Im praktischen Einsatz werden solche Unterstützungsfunktionen unter anderem durch Workflow Management-Systeme (WfMS) angeboten. Basierend auf den Informationen, die zu Prozessbeginn vorliegen, führen sie einen Sachbearbeiter durch den Prozess, indem sie die erlaubten Folgeschritte anhand der aktuellen Informationslage aufzeigen. WfMS arbeiten auf der Grundlage von Prozessmodellen, die durch einen Experten oder automatisch erstellt worden sind, beispielsweise durch Anwendung eines Process Mining-Ansatzes. Zu jedem Prozessschritt wird ein passender Programmbaustein angeboten, über den neue Informationen ein- beziehungsweise ausgegeben oder verarbeitet werden können (Reichert und Dadam 2000).

3.1.1 *TEILGEBIETE DES GESCHÄFTSPROZESSMANAGEMENTS*

GPM ist eine Disziplin mit einem weitgefassten Themenspektrum. Um die Beiträge der Dissertation in dieses Spektrum einordnen zu können, ist eine Aufschlüsselung der Teilgebiete von GPM nützlich. Eine solche Übersicht hierzu liefern Dumas et al. (Dumas 2013):

- Theorien und Methoden zur Modellierung von Geschäftsprozessen (Process Modeling)
- Automatische Erzeugung von Prozessmodellen (Process Discovery)
- Management und Analyse von Prozessen und Prozessmodellen (Process Analysis)
- Verbesserung von Prozessmodellen (Process Redesign)
- Management von Prozessauführungen (Process Automation)

Aus dieser Aufstellung lässt sich ablesen, dass sich die wissenschaftlichen Beiträge zu GPM überwiegend mit den Modellen und deren Betrachtung aus der mathematischen Perspektive beschäftigen. Die Schwerpunkte liegen dabei auf der Verifikation von Modellen beziehungsweise von Prozessen (Motahari-Nezhad et al. 2015).

Den traditionellen GPM-Ansätzen und ihrer Implementierung in WfMS wird unterstellt, dass sie zu restriktiv und nicht anpassbar seien (Aalst et al. 2005). Das führt dazu, dass solche Technologien hauptsächlich für gut strukturierte, wiederholbare Prozesse eingesetzt werden können. Wissensintensive Prozesse lassen sich im Gegensatz dazu nicht durch WfMS unterstützen (Schwarz et al. 2001).

3.1.2 *CASE HANDLING: GPM FÜR SCHWACH STRUKTURIERTE PROZESSE*

Um gezielt auch Prozesse zu unterstützen, die selten in vorab festgelegten Schritten ablaufen, wurde in (Aalst et al. 2005) der Case Handling-Ansatz eingeführt. Im Vergleich zu den bisherigen Beiträgen der GPM-Community handelt es sich dabei um ein Konzept mit einem flexibleren Mechanismus, der beispielsweise das Überspringen von Aktivitäten während eines Prozessablaufs ermöglicht. Diesem Mechanismus wird ein Modell zugrunde gelegt (siehe Abbildung 5), in dem neben den Aktivitäten (A) eines Falls (C) auch die dazugehörigen Dialoge (F) und benötigten Daten (D) spezifiziert werden.

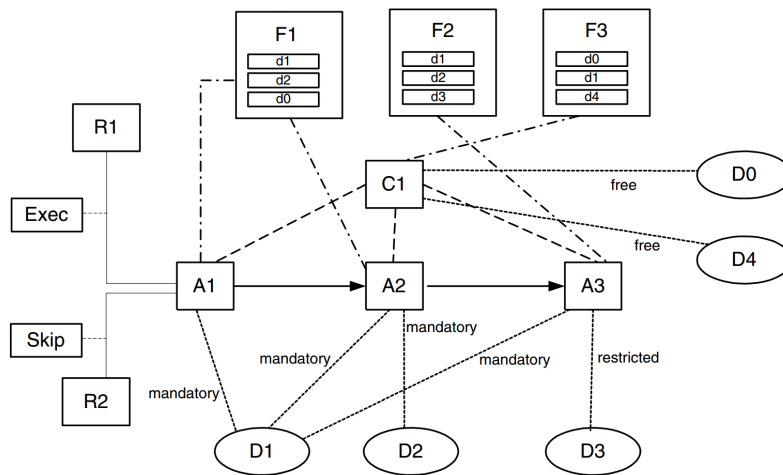


Abbildung 5: Abstraktes Beispiel eines Case Handling-Modells (Aalst et al. 2005)

An der dem Fall C1 zugeordneten Aktivität A1 sind zwei unterschiedliche Rollen (R1, R2) beteiligt. Einer Rolle wohnt die Fähigkeit inne, eine Aktivität auszuführen (Exec) oder zu überspringen (Skip). Der Aktivität wird außerdem ein Dialog F1 zugeordnet. Er zeigt bestimmte Attribute (d1, d2, d0) eines Datenobjekts (D0, D1, D2) an. Attribute von Daten (D0 und D4), die dem Fall frei zugeordnet wurden, können in allen Formularen angezeigt werden. Manche Daten können für die Ausübung mehrerer Aktivitäten notwendig sein (mandatory). Es können aber auch Daten spezifiziert werden, die ausschließlich für eine bestimmte Aktivität verwendet werden dürfen (restricted).

Die Zuordnung von Dialogen, Aktivitäten und Programmbausteinen selbst ist kein neuer Ansatz (Reichert und Dadam 2000). Die Originalität des Case Handlings bezieht sich eher darauf, dass im Modell festgelegt werden kann, dass die Sachbearbeiter nunmehr Aktivitäten überspringen können und dass die Daten präziser in das Modell eingebunden werden können. Beide Elemente bringen eine neue Funktionalität, die von WfMS bis dahin nicht berücksichtigt wurden. Es ist aber anzweifelbar, ob sie flexibel genug sind, um tatsächlich Wissensarbeit beziehungsweise Case Management zu unterstützen. Denn bezogen auf die Klassifikation von (Loeffeler et al. 1998) sind in schwach strukturierten Prozessen die Daten, die beteiligten Personen und der Lösungsweg nicht unbedingt im Vorfeld bekannt. Fehlen manche dieser Informationen, fällt eine Zuordnung schwer, welche Rollen Aktivitäten überspringen dürfen und auf welchem Formular Attribute bestimmter Geschäftsobjekte abgebildet sind.

3.1.3 NOTATIONEN FÜR SCHWACH STRUKTURIERTE PROZESSE

Mit BPMN (Business Process Model and Notation) steht der GPM-Community seit 2008 eine standardisierte Notation zur Verfügung, um Geschäftsprozesse detailliert spezifizieren zu können (Object Management Group 2013). Die in der BPMN enthaltenen Notationselemente bieten aber

mit dem herkömmlichen Sequenz- und Nachrichtenfluss nicht die nötige Flexibilität, um Case Management-Prozesse abbilden zu können (Kurz et al. 2015). Eines der wenigen Notationselemente von BPMN, das sich zur Darstellung nicht strukturierter Prozesse einsetzen lässt, ist der „Ad-Hoc Sub-Process“ (Object Management Group 2013). In diesen Subprozessen sind einige Regeln der BPMN außer Kraft gesetzt. So müssen die darin enthaltenen Aktivitäten beispielsweise nicht miteinander über Sequenzflüsse verbunden werden. Außerdem ist kein Start- und Endereignis nötig. Die Aktivitäten eines Ad-Hoc Sub-Process können parallel, sequentiell und wiederholt ausgeführt werden. Nach jeder Ausführung kann der Teilprozess verlassen werden.

Anhand der besonderen Merkmale von Case Management-Prozessen haben Kurz et al. Anforderungen zusammengetragen, die an eine Notation zur Modellierung von Case Management-Prozessen gestellt werden müssen (Kurz et al. 2015). Ein Beispiel hierfür sind temporallogische Zusammenhänge wie „Vor Beginn der Aktivität A muss (irgendwann) Aktivität B durchgeführt worden sein“. Eine solche Abhängigkeit ist in BPMN nicht darstellbar. Die Sequenzflüsse sind hierfür zu strikt und die Ad-hoc-Teilprozesse sind wiederum zu unverbindlich. Um diese Lücken zu schließen, wurde im Mai 2014 von der Object Management Group die Case Management Model and Notation (CMMN) veröffentlicht (Object Management Group 2014). Sie adressiert fast alle Anforderungen an die Modellierung von Case Management-Prozessen nach (Kurz et al. 2015). Eine beispielhafte Auswahl daraus wird im Folgenden anhand eines Beispiels (siehe Abbildung 6) kurz vorgestellt.

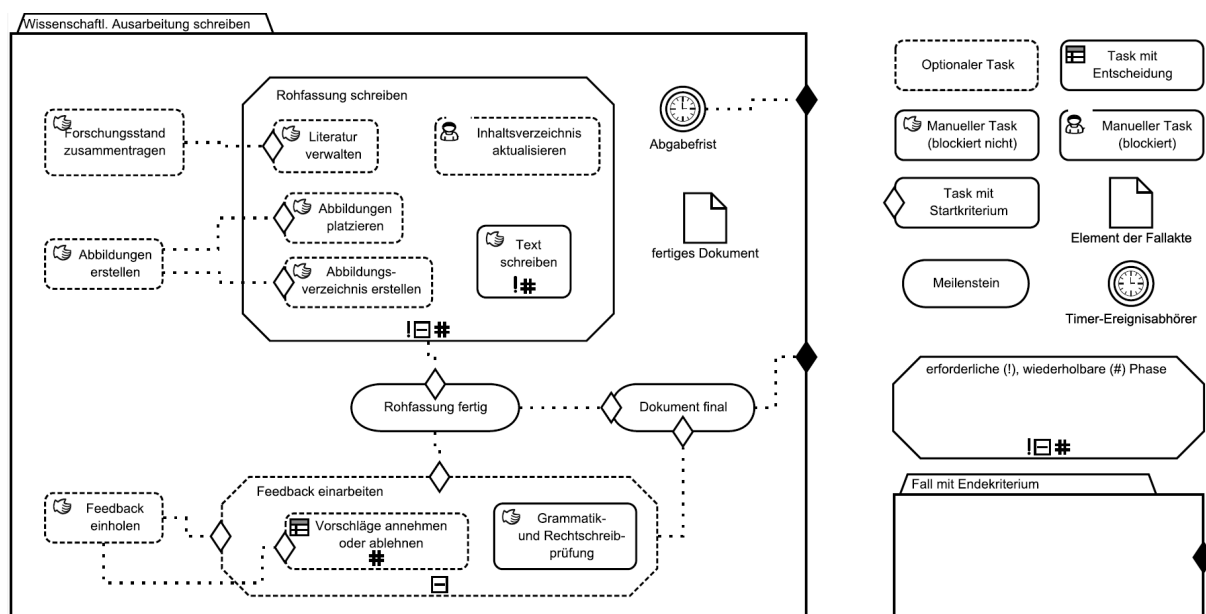


Abbildung 6: CMMN-Modell eines Schreibprozesses, angelehnt an (Kurz et al. 2015); rechts: Legende

Grundsätzlich handelt es sich bei CMMN um eine deklarative Notation (Object Management Group 2014, S. 46). Im Gegensatz zu BPMN hat die Position einer Aktivität im Modell eines Falls (großes Rechteck mit Lasche) per se keine Bedeutung hinsichtlich der Ausführungsreihenfolge. Soll trotzdem eine Reihenfolge festgelegt werden, kann das der Modellierer mithilfe von Start- und Endekriterien (Raute) beschreiben. Ein mit einem Startkriterium versehener Task (abgerundetes Rechteck) kann demnach nur gestartet werden, wenn alle mit der Raute verknüpften Tasks oder Dokumente abgeschlossen sind beziehungsweise vorliegen. Beispielsweise müssen Abbildungen erst erstellt werden, bevor sie platziert und ins Verzeichnis aufgenommen werden können. Falls eine Gruppe von Aktivitäten vor einer anderen Gruppe ausgeführt werden soll, kann dieser Zusammenhang mit Phasen modelliert werden (Achteck), die über Startkriterien miteinander verbunden werden.

Darüber hinaus unterscheidet die Notation zwischen fest eingeplanten Elementen (durchgezogene Umrandung) und optionalen Elementen (gestrichelte Umrandung). Erstere geben dem Modellierer zwar die Möglichkeit, wesentliche Verantwortungen im Prozess zu kennzeichnen, sie können aber trotzdem zur Laufzeit vom Case Manager ignoriert werden. Letztere sind Elemente, deren Ausführung oder Verwendung ausdrücklich im Ermessen des Case Managers liegt. Auf diese Weise wird dem Case Manager die Möglichkeit gegeben, die Elemente eines Falls zur Laufzeit anzupassen. Am Beispiel ist das Schreiben von Text eine zentrale Aufgabe beim Verfassen der Rohfassung. Ob aber ein Abbildungs- oder Inhaltsverzeichnis nötig ist, bleibt im Ermessen des Case Managers. Weil mit diesem Notationselement keine alternativen Verzweigungen und Kontrollflüsse modelliert werden müssen, ist die Abbildung optionaler Elemente verglichen mit traditionellen Notationen wie der BPMN weniger kompliziert.

Tasks können mithilfe von verschiedenen Symbolen typisiert werden. Dadurch kann eine Aufgabe beispielsweise ausdrücklich als „manuell“ deklariert werden. Im Gegensatz zu Tasks, die als „Prozess“ gekennzeichnet wurden, müssen manuelle Tasks vom Case Manager ohne Unterstützung, zum Beispiel durch WfMS, durchgeführt werden.

Wie bereits in Kapitel 2 beschrieben ist der konkrete Lösungsweg zwar in Case Management-Prozessen nicht immer im Vorfeld bekannt. Dennoch gibt es oft konkrete Ziele, die am Ende des Falls erreicht werden sollen. Solche Ziele lassen sich in Form von Endekriterien modellieren (ausgefüllte Raute), die an einen Fall gebunden werden.

3.1.4 ADAPTIVE CASE MANAGEMENT

Etwa seit 2010 haben sich neben van der Aalst auch andere Forscher aus der GPM-Disziplin wie Rychkova, Bider und Swenson, damit auseinandergesetzt, mit welchen Methoden GPM dazu beitragen kann, eine flexiblere Prozessunterstützung zu gewährleisten. Ein wissenschaftliches Forum hierfür bietet der seit 2012 regelmäßig abgehaltene Adaptive Case Management (ACM) Workshop (Business Process Management Workshops 2013; Demey und Panetto 2013; IEEE 2014). Zusätzlich dazu veröffentlichen Autoren aus Wissenschaft und Praxis regelmäßig Fachbücher zu diesem Thema (Swenson 2010a; Swenson 2011; Swenson 2012; Fischer und Miers 2013; Fischer 2015). Eine kurze Einführung in ACM wird an dieser Stelle gegeben, weil sich die Lösungsbausteine der Dissertation gut in die Veröffentlichungen der Adaptive Case Management-Community einsortieren lassen.

Bei Adaptive Case Management (ACM) handelt es sich um einen stark technologieorientierten Ansatz. Demnach sind die Beiträge im Gegensatz zum traditionellen GPM überwiegend von der Praxis geprägt. Die Prozesse, die primärer Gegenstand dieser Technologie sind, haben laut Swenson (2010a) starke kollaborative, dynamische und informationsintensive Anteile. Sie sind von externen Ereignissen getrieben, die stets Reaktionen vom ausführenden Case Manager verlangen. Nach der Klassifikation von (Loeffeler et al. 1998) handelt es sich also um schwach strukturierte Prozesse. Doch auch strukturierte Prozesse sollen gemäß einer Definition von Swenson durch die Anwendung von Adaptive Case Management profitieren:

„Adaptive Case Management (ACM) is information technology that exposes structured and unstructured business information (business data and content) and allows structured (business) and unstructured (social) organizations to execute work (routine and emergent processes) in a secure but transparent manner.“ (Swenson 2010a)

Wie bereits eingangs erwähnt, steht beim traditionellen GPM der Prozess im Mittelpunkt der Forschungsbeiträge. Aus der Prozessperspektive heraus wird beleuchtet, mit welchen Daten und welchen Informationssystemen der Anwender ausgestattet werden muss, um den Prozess ausführen zu können. Im Gegensatz dazu verfolgt der ACM-Ansatz das Ziel, die (Fall-)Daten als festen Bezugspunkt zu wählen. Der Prozess beziehungsweise die Aktivitäten werden dann so gestaltet, dass die Anwender mithilfe der Daten ihre Ziele erreichen.

Die in dieser Dissertation vorgestellten Konzepte adaptieren unter anderem die drei von Pucher vorgestellten Merkmale von ACM (Pucher 2010). Dabei hebt er hervor, dass anstelle der Verwendung vieler isoliert betriebener Anwendungen ein ACM-System das zentrale Werkzeug für

den Case Manager darstellt, mit dem er den gesamten Fall steuern kann. Hierzu bietet das ACM-System – in Ergänzung zu der Prozessunterstützung – auch einen zentralen Knoten für die Verwaltung aller prozessrelevanten Daten. Weiterhin können im Gegensatz zum GPM die Anwender eines ACM-Systems selbst Prozesse definieren oder anpassen. Dabei spielt es keine Rolle, ob die Prozesse imperativ oder deklarativ modelliert wurden. Als drittes Merkmal weist er darauf hin, dass bei traditionellen GPM-Ansätzen das Design und die Analyse von Prozessen abseits der eigentlichen Prozessausführung stattfinden. Bei der Anwendung von ACM sind diese Funktionen hingegen schon während der Fallbearbeitung verfügbar. Dabei versorgt das ACM-System den Anwender mit Wissen über den Prozess, das sich aus den Mustern in dem Verhalten der Anwender selbst speist.

3.1.5 BEWERTUNG UND BEZUG ZUR DISSERTATION

Publikationen wie (Rosing et al. 2014) machen deutlich, dass GPM eine sehr breit gefächerte Disziplin mit einer großen Zahl an wissenschaftlichen Beiträgen darstellt. Es existieren einige produktiv einsetzbare Werkzeuge zur Modellierung und Analyse von strukturierten Prozessen sowie zur Ausführungsunterstützung in Form von WfMS. Als Grundlage für die Unterstützung während der Ausführung ist ein Prozessmodell erforderlich, beschrieben mit standardisierten Notationen wie der BPMN, in dem alle Aktivitäten und deren Abhängigkeiten dargestellt sind.

Um auch schwach strukturierte Prozesse unterstützen zu können, haben Aalst et al. das Case Handling-Paradigma vorgestellt, mit dem mehr Flexibilität in der Ausführung von Aktivitäten geboten werden soll (Aalst et al. 2005). Etwas später wurde in der Literatur aber das Maß an Flexibilität, das durch dieses Paradigma erzielt werden kann, nur als „unverfänglich“ (van der Aalst et al. 2009) bewertet. Das liegt mitunter daran, dass zwar die Aktivitäten abstrakter definiert werden können als in einem Workflow, aber darüber hinaus das Modell nur insofern flexibel ist, als dass Aktivitäten übersprungen werden können. Diese Aktivitäten müssen aber dennoch schon im Vorfeld festgelegt werden. Das grundlegende Konzept eines „imperativen“ Prozessmodells im Sinne eines vorab festgelegten, zusammenhängenden Kontrollflusses wird beim Case Handling nicht aufgegeben.

Den Argumenten in (van der Aalst et al. 2009) und (Rychkova 2013) folgend, werden solche imperativen Modelle nicht die nötige Flexibilität bereitstellen, die im Case Management benötigt wird. Im Gegensatz dazu schlagen van der Aalst und Rychkova vor, mithilfe von deklarativen Notationen festzulegen, in welchen Grenzen sich ein Fall bewegen darf beziehungsweise welche Rahmenbedingung dabei eingehalten werden müssen. In diesem Sinne sind während der Ausführung alle Aktivitäten erlaubt, die nicht ausdrücklich durch das Modell verboten wurden.

Diesen Ansatz verfolgt auch der CMMN-Standard. Dadurch erfüllen die dort definierten Elemente und Konzepte nach (Kurz et al. 2015) die meisten Anforderungen an die Beschreibung von Case Management-Prozessen. Trotzdem handelt es sich bei CMMN nur um eine Notation zur *Beschreibung* und *Kommunikation* von Case Management-Prozessen. Eine *Methode* zur Unterstützung eines Case Managers bietet sie nicht.

Zwar gibt es bereits positive Erfahrungen mit der Anwendung deklarativer Modelle zur Beschreibung von Geschäftsprozessen, aber die Praxisreife der damit verbundenen Konzepte ist zum aktuellen Zeitpunkt noch nicht gegeben (Reijers et al. 2013).

Die Frage, wie die Lösungsbestandteile des adCM mit Adaptive Case Management in Bezug stehen, wird in Kapitel 7.7.4 beantwortet. Dort wird gezeigt, an welcher Position sich adCM in der ACM-Referenzarchitektur von (Pucher 2010) einordnen lässt.

3.2 PROCESS MINING

Wie noch in Kapitel 4.2 gezeigt werden wird, orientieren sich die wesentlichen Lösungsbausteine und damit auch die Gliederung der Arbeit an den im Process Mining verwendeten Begriffen und grundlegenden Vorgehensweisen. Deshalb erfolgt in diesem Abschnitt eine kurze Einführung der zentralen Konzepte des Process Mining. Er schließt mit einer Bewertung des Forschungsstands.

3.2.1 ÜBERBLICK

Process Mining ist eine Forschungsdisziplin zur Erzeugung von Modellen über Geschäftsprozesse auf der Grundlage von Ereignisprotokollen, die durch Softwaresysteme erzeugt werden. Im Fokus der Forschungsbeiträge stehen die Zusammenhänge zwischen den Ereignisprotokollen und den Prozessmodellen (siehe Abbildung 7, unten). Dabei wird zwischen drei unterschiedlichen sogenannten „Typen“ von Process Mining unterschieden: **Erkennung** („Discovery“), **Übereinstimmung** („Conformance“) und **Erweiterung** („Enhancement“). Beiträge, die sich auf das Discovery spezialisieren, beschäftigen sich mit der Erzeugung von Prozessmodellen aus den Ereignisprotokollen. Der Schwerpunkt liegt dabei zwar auf imperativen Kontrollfluss- und Zustandsübergangsmodellen wie Petrinetzen, Transitionssystemen oder BPMN-Diagrammen. Es gibt aber auch Arbeiten wie (Song und van der Aalst 2008), die sich der Erhebung anderer Sichten widmen, beispielsweise sozialer Strukturen („Organizational Mining“).

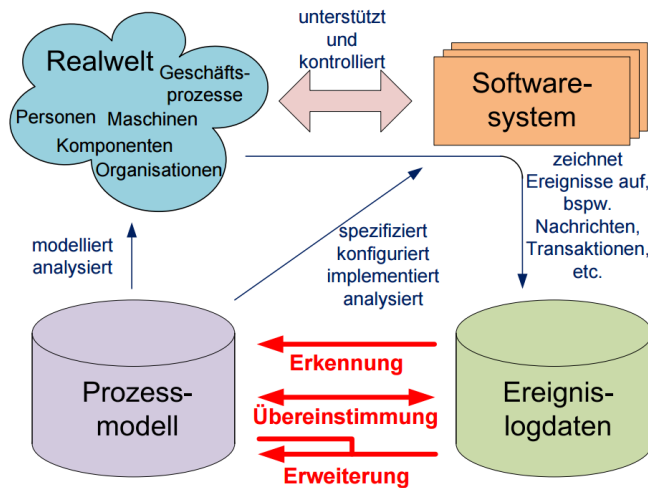


Abbildung 7: Die drei Typen des Process Mining (van der Aalst et al. 2012)

3.2.2 HERAUSFORDERUNGEN BEIM MONITORING UND DISCOVERY

Unter Berücksichtigung der Zielstellung der Dissertation werden im weiteren Verlauf nur die Aspekte des Monitorings, also der Ereignisaufzeichnung (Realwelt → Ereignislogdaten), und des Discovery (Ereignislogdaten → Prozessmodell) vertieft.

Nach (van der Aalst 2011) sind Ereignisse wesentliche Informationsträger für das Process Mining. Er hebt diesen Aspekt deshalb besonders hervor, weil solche Daten üblicherweise Nebenprodukte in Informationssystemen seien, zum Beispiel für die Fehleranalyse. Für die Anwendung von Methoden des Process Mining muss diesen Ereignisdaten allerdings eine besondere Aufmerksamkeit zuteilwerden. Zu diesem Zweck stellt van der Aalst die folgenden vier Kriterien zur Beurteilung der Ereignisdatenqualität auf:

Belastbarkeit: Ereignisse müssen tatsächlich mit den angegebenen Attributen stattgefunden haben.

Vollständigkeit: Wurde eine bestimmte Abstraktionsebene gewählt, auf der die Aktivitäten protokolliert werden, so muss das Protokoll auf dieser Ebene vollständig den Prozess abbilden.

Semantik: Die Daten sollen idealerweise mit einer Semantik versehen werden, zum Beispiel in Form eines Bezugs zu einem Begriff, der in einer Ontologie enthalten ist.

Datenschutz- und Datensicherheit: Die Protokollierung von Benutzerinteraktionen stellt je nach Gesetzeslage einen mehr oder weniger starken Eingriff in die Privatsphäre des Benutzers dar. Gegebenenfalls müssen die erhobenen Daten anonymisiert oder pseudonymisiert werden. In jedem Fall sollten die Benutzer über die Protokollierung informiert sein.

Entlang dieser Kriterien wird die Qualität der Protokolle nach (van der Aalst et al. 2012) mit einer fünfstufigen Skala klassifiziert. Für die Effektivität von Process Mining-Methoden wird laut van der Aalst eine Datenqualität von mindestens Stufe 3 vorausgesetzt. Weil die Qualität der Ereignisdaten im Kapitel 5 eine hervorgehobene Rolle spielt, werden diese Stufen an dieser Stelle kurz zusammengefasst:

1. Stufe: Es ist unklar, ob die Ereignisse tatsächlich mit dem Prozess zusammenhängen. Es ist ebenso unklar, ob die Ereignisse vollständig aufgezeichnet werden. Beispiele: Protokolle, die von Hand geführt oder aus verschiedenen unstrukturierten Notizen zusammengetragen werden.
2. Stufe: Ereignisse werden zwar automatisch von Informationssystemen aufgezeichnet, aber diese Aufzeichnung erfolgt nicht systematisch. Zudem können die Systeme umgangen werden, sodass nicht mit Sicherheit festgestellt werden kann, ob die Protokolle vollständig sind. Beispiele: Fehlerprotokolle, die ursprünglich für die Wartung der Informationssysteme geführt werden.
3. Stufe: Ereignisse werden automatisch und zuverlässig aufgezeichnet. Das heißt, sie bilden mit hoher Wahrscheinlichkeit den Prozess korrekt ab. Beispiele: Protokolle aus Informationssystemen, aus denen sich unmittelbar die Zustandsveränderungen der Geschäftsobjekte während des Prozesses ablesen lassen wie Enterprise Resource Planning-Systeme oder Customer-Relationship-Systeme.
4. Stufe: Ereignisse werden automatisch, systematisch und zuverlässig aufgezeichnet. Dadurch sind sie vollständig und bilden den Prozess korrekt ab. Sie erlauben darüber hinaus die Unterscheidung einzelner Fälle und die Abbildung von Aktivitäten. Beispiele: Protokolle von WfMS
5. Stufe: Ereignisse werden wie in Stufe 4 aufgezeichnet, sind darüber hinaus aber klar definiert, mithilfe einer Ontologie mit einer Semantik versehen und berücksichtigen auch Aspekte des Datenschutzes und der Datensicherheit. Beispiele: Semantisch verknüpfte Protokolle von GPM-Systemen

Um möglichst aussagekräftige und lesbare Prozessmodelle zu erhalten, sollten die Ereignisse möglichst gut durch Modellelemente darstellbar sein. Dies kann am besten dadurch erzielt werden, dass die Ereignisse zumindest auf dem gleichen Abstraktionsniveau beschrieben sind. Aber noch viel wichtiger ist nach (van der Aalst 2011) die Kohärenz zwischen den Ereignissen und den Modellelementen bei der Überprüfung, ob ein gegebenes Protokoll dem Modell entspricht (**Conformance Checking**). Ohne eine solche Abbildung kann nach van der Aalst keine sinnvolle Überprüfung stattfinden.

Beim Process Mining ist laut (van der Aalst 2011) eine Metapher zu beachten. Demnach haben Prozessmodelle und Landkarten bestimmte Gemeinsamkeiten: Es gibt nicht das *eine* Modell als einzig wahres Abbild der Realität. Stattdessen gibt es viele unterschiedliche Abstraktionsebenen und Schwerpunkte, die je nach Zielgruppe mehr oder weniger zweckmäßig sind. Deswegen muss festgelegt werden, für welche Zielgruppe ein Modell ermittelt wird und bei Bedarf müssen zusätzliche Sichten zu dem Modell ermittelt werden.

Viele Process Mining Forscher gehen davon aus, dass sich strukturierte Prozesse bereits während der Analyse verändern können (van der Aalst et al. 2012). Die dadurch entstehende Dynamik führt dazu, dass Process Mining kein einmaliger Vorgang ist, sondern in regelmäßigen Abständen wiederholt werden muss, um die Prozessmodelle aktuell zu halten, beziehungsweise die Konformität der Ausführungsvarianten nicht aus den Augen zu verlieren. Außerdem ist kontinuierliches Process Mining die Voraussetzung für die Bereitstellung von Echtzeit-Informationen zu den laufenden Prozessen.

3.2.3 BEWERTUNG UND BEZUG ZUR DISSERTATION

Die Diskussion der Anwendbarkeit des Process Mining auf schwach strukturierte Prozesse wird in (van der Aalst 2011) diskutiert. Dort unterscheidet Aalst zwischen sogenannten „Lasagna-Processes“ einerseits, die bei der Anwendung von Discovery-Algorithmen auf strukturierte Prozesse entstehen, und den sogenannten „Spaghetti-Processes“ andererseits, die bei der Anwendung auf schwach strukturierte Prozesse entstehen (siehe Abbildung 8).

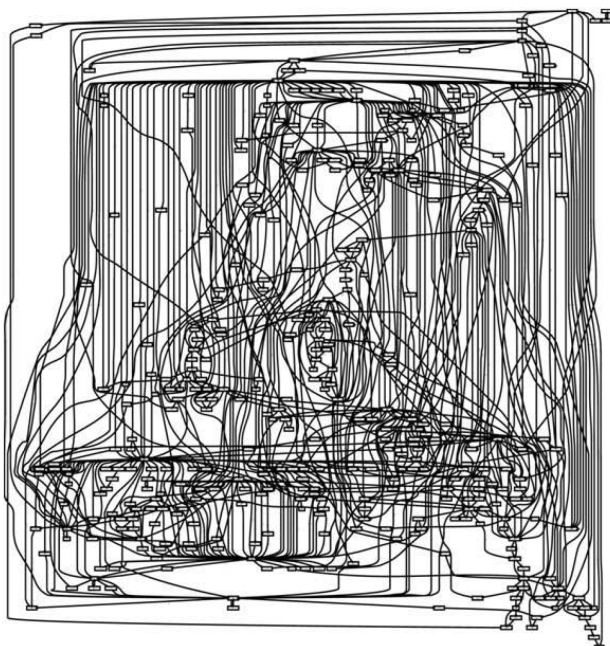


Abbildung 8: Spaghetti-Prozess nach (van der Aalst 2011)

Die beiden Arten von Modellen sind jeweils nach ihrem äußeren Erscheinungsbild benannt. Demzufolge sind die letztgenannten aufgrund ihrer Komplexität nicht als Dokumentation oder Kommunikationsgrundlage geeignet. Auch der Einsatz solcher Modelle in WfMS ist aufgrund der vielen Alternativen nicht sinnvoll.

Aalst empfiehlt daher eine leicht abgewandelte Verfahrensweise des Process Mining, deren grundlegender Aufbau in den Lösungsbausteinen der Dissertation aufgegriffen und als Schablone zur Einordnung der Beiträge eingesetzt wird (siehe Kapitel 4.2). Eine Diskussion konkreter Discovery-Algorithmen und mit welchem Erfolg sie sich auf schwach strukturierte Prozesse anwenden lassen, ist Gegenstand der Kapitel 6.2 beziehungsweise 6.4.

3.3 WISSENSBASIERTE SYSTEME

Ein wesentlicher Beitrag dieser Dissertation ist die Unterstützung von Case Managern beim Planen zukünftiger Prozessschritte mithilfe von Vorschlägen, die sich auf Muster aus vergangenen Fällen beziehen. Eine solche Funktion lässt sich in den Verantwortungsbereich von wissensbasierten Systemen (auch: Expertensysteme) einordnen. Deshalb folgt in diesem Unterkapitel eine kurze Einführung in das Case-based Reasoning (CBR), dessen grundlegendes Konzept sehr den in dieser Dissertation vorgestellten Beiträgen ähnelt. Daran anknüpfend wird mit dem Case-Based Planning-Ansatz (CBP) auf eine bestimmte Art von CBR-Systemen eingegangen, die sich gezielt mit der Unterstützung von Planungsaktivitäten befassen. Das Unterkapitel schließt mit einer Bewertung dieser Forschungsbeiträge unter Berücksichtigung der Zielstellung dieser Arbeit. Außerdem wird in der Bewertung dargelegt, welche Ansätze aus den Beiträgen adaptiert werden.

3.3.1 CASE BASED REASONING

Bei CBR handelt es sich im Wesentlichen um eine Methode, mit der zu einem aktuellen Fall unter Einbeziehung einer Falldatenbank vergleichbare Fälle gesucht und Lösungen aus diesen Fällen vorgeschlagen werden. Nach Abschluss des Falls wird er zusammen mit Informationen über seine „Lösung“ ebenfalls in der Falldatenbank (auch: Fallbasis) zurückgespeichert. Auf diese Weise entsteht dabei eine kontinuierlich weiterentwickelte Wissensbasis aus Problemen und Lösungen. Wie später in Kapitel 4.3.4 noch gezeigt wird, adaptiert adCM diese grundlegende Systematik.

Nach Aamodt und Plaza sind CBR-Systeme wissensbasierte Systeme und lassen sich somit der Disziplin der künstlichen Intelligenz zuordnen (Aamodt und Plaza 1994). Unter dem

übergeordneten Ziel, computergestützt Probleme zu lösen, beschreibt CBR eine Methode für Expertensysteme, die je nach Anwendungsfall verfeinert werden muss.

Aamodt und Plaza definieren CBR als einen Zyklus, bestehend aus vier sequentiellen Schritten (siehe Abbildung 9). Der Zyklus startet damit, dass dem CBR-System ein neues, zu lösendes Problem präsentiert wird. Es muss in einer bestimmten Form (Case Model) vorliegen, das vom System gelesen und verstanden werden kann. Hierzu müssen gegebenenfalls die charakteristischen Merkmale des Falls im Allgemeinen beziehungsweise des zu lösenden Problems im Speziellen extrahiert werden.

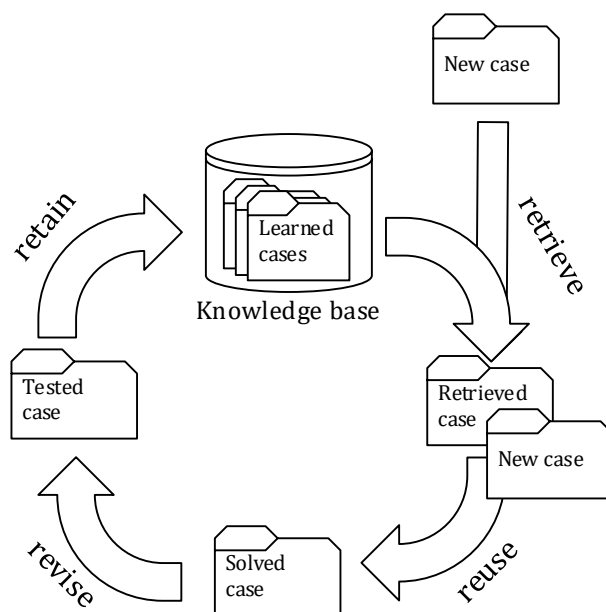


Abbildung 9: Zyklus des Case-based Reasoning nach (Aamodt und Plaza 1994)

Die vier Schritte – wegen ihres gemeinsamen Anfangsbuchstabens auch oft als „the 4R“ bezeichnet – lauten nach (Aamodt und Plaza 1994) wie folgt:

Retrieve: Entnehme aus der Wissensdatenbank den ähnlichsten Fall beziehungsweise den Fall mit der ähnlichsten Problembeschreibung. Je nach Implementierung können das auch mehrere passende Fälle sein.

Reuse: Schlage die Lösungen aus vergleichbaren Fällen vor und verwende sie wieder, um das neue Problem zu lösen.

Revise: Überprüfe, ob die vorgeschlagene Lösung tatsächlich geholfen hat, um das neue Problem zu lösen.

Retain: Ergänze die Wissensbasis um die (im Regelfall auf die neue Problemstellung angepasste) Lösung zusammen mit der gelösten Problemstellung.

Die Schritte werden von Aamodt und Plaza durch einen Aufgaben-Baum weiter verfeinert, bis den Blättern konkrete Methoden zugeordnet werden können. Demnach untergliedert sich beispielsweise der Schritt „retrieve“ in die Teilaufgaben „identify features“, „search“, „initially match“ und „select“. Bei diesen Aufgaben geht es darum, Schlagwörter zu finden, die den aktuellen Fall möglichst gut beschreiben, alte Fälle aus der Falldatenbank zu laden, mithilfe der Schlagwörter diese Fälle zu filtern und anschließend eine Auswahl zu treffen, welche der herausgefilterten Fälle dem aktuellen Fall am ähnlichsten sind. Das Filtern („initially match“) wird wiederum im Detail als ein Prozess beschrieben, in dem die Schlagwörter als Indizes für den Zugriff auf die Falldatenbank verwendet werden.

3.3.2 CASE BASED PLANNING

Nach (Spalzzi 2001) ist Case Based Planning (CBP) ein auf CBR aufbauendes Forschungsgebiet. Es beschäftigt sich mit Methoden zur Lösung von Planungsproblemen mithilfe der Wiederverwendung von Plänen, die in der Vergangenheit erfolgreich waren. Der Einsatz von CBP-Systemen soll laut Spalzzi dabei helfen, die Effizienz von Planungsprozessen zu verbessern.

Die wesentlichen Schritte nach (Spalzzi 2001) decken sich weitestgehend mit den „4 Rs“ aus dem CBR und bilden ebenfalls einen Zyklus entlang der Fallbearbeitung. Im Unterschied zum CBR beziehen sie sich aber jeweils auf den Plan:

Plan Memory Representation: Festlegen, welche Informationen zu einem Plan gehören und wie sie gespeichert werden sollen

Plan Retrieval: Methoden zur Identifizierung von Plänen, die in der Vergangenheit ein ähnliches Problem gelöst haben

Plan Reuse: Methoden zur Übernahme oder Anpassung eines passenden Plans

Plan Revision: Methoden zur Überprüfung, ob der adaptierte Plan tatsächlich zum Erfolg geführt hat (inklusive Korrektur des Plans)

Plan Retention: Methode zur Ablage des Plans in einer Form, dass er wieder für zukünftige Problemstellungen herangezogen werden kann

Die konkreten Unterschiede zum CBR zeigen sich erst bei einer tiefergehenden Betrachtung wie zum Beispiel der Dekomposition der übergeordneten Schritte in einem Task-Baum. Beim Matching werden beispielsweise konkrete Methoden benannt, mit denen ähnliche Pläne oder Teile daraus gesucht werden. In der Literatur wird dabei zwischen dem assoziativen, dem hierarchischen und dem modellbasierten Matching sowie einer Mischform aus diesen Ansätzen unterschieden. Im weiteren Verlauf wird auf den hierarchischen Ansatz genauer eingegangen, weil er einem der zentralen Konzepte des adCM-Metamodells sehr ähnlich ist.

Im hierarchischen Ansatz werden abstrakt formulierte Ziele sukzessive zergliedert, bis sie einer vorhandenen konkreten Lösung zugeordnet werden können. Diese Lösungen können dann rückwärts entlang des Baumes wieder zu abstrakteren Lösungen zusammengefasst werden (Yang 1997). Auf diese Weise können nicht nur konkrete Lösungen für abstrakte Probleme gefunden werden, sondern auch umgekehrt Lösungen so abstrahiert werden, dass sie in zukünftigen Fällen angewendet werden können.

3.3.3 BEWERTUNG UND BEZUG ZUR DISSERTATION

Die Anwendung der CBR-Methode setzt voraus, dass die Struktur der Datenbank, in denen die Fälle abgelegt werden, unter anderem zwei grundlegende Anforderungen erfüllt: Erstens müssen die charakteristischen Merkmale (Features) des Problems bekannt sein, das in einem Fall gelöst wurde. Dabei handelt es sich nach (Beierle und Kern-Isberner 2008) um eine essentielle Voraussetzung, die in der Praxis aber selten gegeben und schwierig zu gewährleisten ist (Mark et al. 1996). Zweitens müssen zu jedem Fall die verwendeten Lösungen festgehalten und ebenfalls so strukturiert werden, dass sie maschinenlesbar sind.

Weil die Suche nach ähnlichen Fällen in der gesamten Fallbasis ohne entsprechende Verschlagwortung der Fälle eine umfangreiche Aufgabe ist, bietet sich nach (Spalazzi 2001) der CBP-Ansatz an. Hier sind lediglich die Pläne Gegenstand der Methode, nicht die gesamte Fallakte. Aber auch weil die Zielstellung der Dissertation auf die Unterstützung von Planungsaktivitäten fokussiert, erscheinen die Konzepte des CBP-Ansatzes vielversprechend.

Zusammengenommen adaptieren die in dieser Dissertation vorgestellten Lösungsbeiträge drei wesentliche Konzepte des CBR und des CBP:

- Der Einsatz einer hierarchischen Datenstruktur für den Plan eines Case Managers, mit deren Hilfe die abstrakten Ziele zu konkreten Handlungsschritten dekomponiert werden können (siehe Kapitel 4.3.1)

- Der Einsatz einer stetig wachsende Fallbasis (mit inhaltlichem Schwerpunkt auf die Dokumentation des jeweiligen Plans), die im Rahmen der Planung zukünftiger Handlungen eines Case Managers zur Erzeugung von Lösungsvorschlägen herangezogen wird (siehe Kapitel 6.3.2)
- Der Einsatz von mehrstufigen Filtermechanismen beim Vorschlagen passender Lösungsalternativen (siehe Kapitel 7.5)

3.4 ZUSAMMENFASSUNG

GPM ist eine Disziplin, die sich im Kern mit Konzepten, Methoden und Werkzeugen zur IT-Unterstützung von Geschäftsprozessen befasst. Die Beiträge der GPM, von der Analyse bis zur operativen Unterstützung durch WfMS, setzen dabei jeweils auf einem zentralen Element auf: ein kontroll- oder datenflussorientiertes Prozessmodell. Diese Modelle lassen sich für strukturierte Prozesse gut erstellen, bieten aber für schwach strukturierte Prozesse nicht den erforderlichen Handlungsspielraum. Deshalb haben sich GPM-Forscher mit alternativen Notationen und Methoden beschäftigt. Ein Beispiel ist das Case Handling-Paradigma, nach dem unter anderem eine höhere Abstraktionsebene bei der Beschreibung von Aktivitäten sowie die Möglichkeit zum Überspringen von Aktivitäten für mehr Flexibilität während der Ausführung sorgen sollen. Doch auch dieses Paradigma bleibt dem grundlegenden Gedanken eines flussorientierten Modells verhaftet. Welche Aktivitäten übersprungen werden dürfen, liegt nach wie vor im Ermessen desjenigen, der den Prozess verantwortet und nicht beim Ausführenden. Genau diese Flexibilität ist aber in der Wissensarbeit erforderlich, deren Prozesse emergent sind und wo Prozesskonformität die Kreativität des Ausführenden einschränkt.

Mit der Veröffentlichung des CMMN-Standards ist eine Notation entstanden, die die Verantwortung für die Ausgestaltung eines Prozesses schwerpunktmäßig vom Prozessverantwortlichen auf den Wissensarbeiter überträgt. Entgegen der imperativen, flussorientierten Notationen wie der BPMN handelt es sich bei CMMN um eine deklarative Sprache. Demnach ist bei der Ausführung eines Prozesses prinzipiell alles erlaubt, was nicht ausdrücklich durch Einsatz bestimmter Modellelemente verboten ist beziehungsweise erfordert wird. Dieser Grundgedanke sorgt für die in wissensintensiven Prozessen allgemein und im Case Management insbesondere benötigte Flexibilität und wird in dieser Dissertation aufgegriffen.

Der Paradigmenwechsel des Adaptive Case Management besagt, dass das zentrale Moment in der Unterstützung von wissensintensiven Geschäftsprozessen nicht in einem Prozessmodell liegt, sondern dass das Prozessmodell vielmehr nur eines von vielen organisatorischen Werkzeugen ist,

mit denen Wissensarbeiter ihre Arbeit um den Fall herum strukturieren. In der praktischen Umsetzung bedeutet das, dass ACM-Systeme als Kern kein Prozessmodell besitzen, sondern die Fallakte mit allen für die Ausführung des Prozesses benötigten Informationen. Dieses Paradigma wird ebenfalls in dieser Dissertation adaptiert.

Das Process Mining ist ein Teilgebiet des GPM und überträgt die Konzepte des Data Mining auf die Analyse von Geschäftsprozessen. Zu diesem Zweck bietet es mit den drei Typen „Discovery“, „Conformance“ und „Enhancement“ einen Rahmen, in den sich die einzelnen Beiträge einordnen lassen. Wenngleich sich die meisten Beiträge im Process Mining auf strukturierte Prozesse konzentrieren, ist das generelle Vorgehen auch auf schwach strukturierte Prozesse anwendbar. Die wesentlichen Schritte des Process Mining, vom Aufzeichnen der Benutzerinteraktionen über deren Analyse bis zur operativen Unterstützung, werden deshalb zur Gliederung der in dieser Dissertation beschriebenen Lösungsbeiträge verwendet.

Abseits der oben dargelegten Disziplinen, die sich schwerpunktmäßig in die Wissenschaft der Wirtschaftsinformatik einordnen lassen, streifen die Beiträge der Dissertation auch Themen, die der künstlichen Intelligenz als Disziplin im Allgemeinen und wissensbasierten Systemen im Speziellen zuzuschreiben sind. Die Unterstützung von Fällen auf Grundlage von Erfahrungen, die in einer Fallbasis gesammelt werden, ist Gegenstand des Case Based Reasoning. Dabei handelt es sich um eine Methode, die entlang des Lebenszyklus eines Falls Teilschritte definiert, mit deren Hilfe die wesentlichen Merkmale eines Falls identifiziert, vergleichbare Fälle in der Fallbasis gefunden und deren Lösungen bezogen auf das aktuelle Problem vorgeschlagen werden. Das Case Based Planning stellt eine Konkretisierung dieses Ansatzes auf die Unterstützung von Planungsaktivitäten dar. Bezugnehmend auf die Zielstellung der Dissertation war eine Betrachtung dessen erforderlich. Die Betrachtung hat ergeben, dass insbesondere das Konzept einer hierarchischen Dekomposition der Eigenschaften eines Plans den Lösungsbeiträgen des adCM ähnelt. Zudem werden der CBR- beziehungsweise CBP-Zyklus allgemein sowie das dabei eingesetzte mehrstufige Vorgehen in dieser Dissertation angewendet.

Wie die vorgestellten Ansätze konkret auf die Lösungsbeiträge des Agenda-driven Case Management einwirken, wird in den nachfolgenden Kapiteln beleuchtet.

4 AGENDA-DRIVEN CASE MANAGEMENT

In diesem Kapitel wird die einleitende Zielstellung aus Kapitel 1.2 verfeinert. Hierzu werden konkrete Anforderungen an den konzeptionellen Beitrag der Dissertation gestellt. Darauf aufbauend werden die einzelnen Lösungsbausteine der Dissertation in einem Schaubild dargestellt und kurz zusammengefasst. Das Schaubild gibt zugleich Aufschluss über die Struktur des Dokuments und wird in diesem Dokument jeweils zu Beginn eines Kapitels zur Orientierung mit angegeben.

4.1 ANFORDERUNGEN AN DIE KONZEPTE DES AGENDA-DRIVEN CASE MANAGEMENT

Wie bereits einleitend festgelegt, widmet sich die Dissertation dem übergeordneten Ziel, **Konzepte zur IT-Unterstützung von Wissensarbeitern** bei der Ausübung wissensintensiver Prozesse zu erarbeiten. Dabei soll ein Schwerpunkt auf Aktivitäten zur Planung und Organisation von Fällen im Case Management gelegt werden. Um dieses Ziel zu konkretisieren, werden nachfolgend **Anforderungen an diese Konzepte** aufgezählt. Um sie thematisch zu gruppieren, erfolgt die Aufzählung entlang der einzelnen Aktivitäten. An vorderster Stelle steht dabei die Unterstützung von Planungsaktivitäten. Ergänzend dazu werden Anforderungen an die Unterstützung komplementärer Aktivitäten gestellt, die gemäß den Ausführungen in Kapitel 2.3 für verschiedene Phasen im Case Management-Prozess ebenfalls benötigt werden.

4.1.1 UNTERSTÜTZUNG DES PLANENS

Gemäß der Definition aus Kapitel 2.2.2 handelt es sich beim Planen um das gedankliche Fortentwickeln eines mentalen Modells über das zukünftige Handeln. Im Case Management findet diese Aktivität schwerpunktmäßig im Rahmen der Zielvereinbarungsphase (*Planning*) statt. Um den Unterstützungsbedarf des Case Managers für diese Phase zu konkretisieren, werden zwei Anforderungen an die konzeptionellen Beiträge der Dissertation gestellt.

Anforderung 1: Unterstützung bei der strukturierten Externalisierung des Plans

Das adCM soll ein Konzept bereitstellen, mit dem der Case Manager seinen **gedanklich vorstrukturierten Plan externalisieren** kann. Weil ein solcher Plan definitionsgemäß einer

ständigen Fortentwicklung unterlegen ist, soll das Konzept eine **angemessene Flexibilität bieten**. Das heißt, der externalisierte Plan muss jederzeit umstrukturierbar sein können.

Anforderung 2: Unterstützung bei der Fortentwicklung des Plans

Der Schwerpunkt dieser Dissertation liegt darin, den Case Manager bei der Identifikation zukünftiger Schritte zur Bearbeitung des Falls zu unterstützen. Das adCM soll deshalb Konzepte und Methoden zusammenstellen, mit denen dem Case Manager automatisch **Vorschläge für die Weiterentwicklung des Plans** unterbreitet werden können.

Eine grundsätzliche Gefahr bei der Bereitstellung von Informationen ist, dass die Wissensarbeiter „quantitativ über- und qualitativ unterinformiert werden“ (Weidt 2004). Dieses Problem wurde bereits in den 1980er Jahren von John Naisbitt formuliert: „We are drowning in information, but starved for knowledge“ (Naisbitt 1984). Um das Problem zu umgehen, soll das Konzept einen **Schwerpunkt auf die Effizienz der Unterstützung** legen. Das heißt, die Qualität der Vorschläge soll Vorrang haben vor der Quantität. Zudem soll der Zeitpunkt der Vorschläge so gewählt werden, dass der Case Manager durch die Vorschläge **nicht in seiner Arbeit unterbrochen** wird.

4.1.2 UNTERSTÜTZUNG DES RECHERCHIERENS

In Kapitel 2.2.2.2 wurde anhand eines Beispiels motiviert, welche Herausforderungen für die IT-Unterstützung von Recherchetätigkeiten im Kontext vielfältiger und vorab nicht fest eingrenzbarer Informationsquellen entstehen. Auf Grundlage dieser Herausforderungen wird die folgende Anforderung an das adCM gestellt.

Anforderung 3: Unterstützung beim Zugriff auf heterogene Informationsquellen

Um die Recherchetätigkeiten eines Case Managers zu unterstützen, soll das Agenda-driven Case Management ein Konzept zur **Vereinfachung des Datenzugriffs** auf vielfältige Datenquellen enthalten.

4.1.3 UNTERSTÜTZUNG DES ARCHIVIERENS UND STRUKTURIERENS

Die Problematik der vielfältigen Datenquellen während der Recherche setzt sich auch bei Aktivitäten zur strukturierten Ablage dieser Daten fort (siehe Kapitel 2.2.2.3). In diesem Zusammenhang werden die beiden folgenden Anforderungen an das adCM gestellt.

Anforderung 4: Unterstützung bei der Ablage von heterogenen Daten

Das adCM soll ein Lösungskonzept beschreiben, das die zentrale Ablage von Informationen aus vielfältigen Datenquellen in einer Art Fallakte ermöglicht. Wesentliche Qualitätsmerkmale dieser Ablage sind die Verfügbarkeit und Performanz. Denn wie in den Kapiteln 2.3 und 3.1.4 motiviert, bilden die Informationen in der Fallakte die Grundlage für die konkrete Ausgestaltung des Prozesses durch den Case Manager.

Anforderung 5: Unterstützung beim Strukturieren der Falldaten

Um die Daten untereinander und mit den geplanten Zielen in Bezug setzen zu können, soll das adCM-Konzept eine Funktion ausgestalten, die das Pflegen von Beziehungen zwischen diesen Elementen ermöglicht.

4.1.4 UNTERSTÜTZUNG DES REFLEKTIERENS

Im Kapitel 2.2.2.4 wurde einführend beschrieben, welche Funktionen vom Wissensarbeiter beim Reflektieren der zusammengetragenen Daten benötigt werden. Ein wesentlicher Bestandteil war das Annotieren von Elementen in der Fallakte mit Meta-Daten wie zum Beispiel Kommentaren oder Schlagwörtern. Um den Case Manager dabei zu unterstützen, solche Annotationen anzuwenden und mit ihnen zu arbeiten, werden in diesem Abschnitt in Anlehnung an (Benner-Wickner et al. 2013) drei Anforderungen an das adCM-Konzept formuliert.

Anforderung 6: Annotieren trotz fehlenden Schreibzugriffs

In allen gängigen Dateiformaten sind Meta-Daten Bestandteil der Datei selbst. Hat der Case Manager darauf keine Schreibrechte, kann er keine Annotationen hinzufügen. Das ist immer dann ein Problem, wenn die Dateien extern bereitgestellt werden oder wenn der Case Manager keine Freigabe für die intern bereitgestellten Daten hat. Wenn die Datenquellen vorab bekannt sind, können die entsprechenden Schreibrechte angefragt oder eingeräumt werden. Weil aber diese Voraussetzung im Case Management nicht gegeben ist, soll das adCM-Konzept eine Lösung bereitstellen, mit der alle Daten in der Fallakte annotiert werden können, selbst wenn der Case Manager nicht überall Schreibrechte besitzt.

Anforderung 7: Annotieren jeglicher Dateiformate

Die Vielfalt der benötigten Daten ist auch insofern ein Problem, als dass sie – wenn überhaupt – jeweils auf ganz unterschiedliche Weise das Annotieren unterstützen. Das bedeutet, dass es keine gemeinsame Schnittmenge an Annotationsfunktionen gibt. Zudem sind die jeweils

zugrundeliegenden Dateiformate nur darauf ausgelegt, die Annotationsfunktionen der dazugehörigen Anwendung zu speichern. Die Aufnahme weniger üblicher Funktionen wie das Zuordnen von Schlagwörtern ist deshalb nicht einheitlich möglich. Das adCM soll eine Lösung enthalten, wie ein übergreifend festgelegter Satz an Funktionen zum Annotieren von Metadaten unabhängig vom Dateiformat der jeweiligen Datenquelle umgesetzt werden kann.

Anforderung 8: Annotieren mit einer einheitlichen Benutzerschnittstelle

Aufgrund der oben angesprochenen Vielfalt der Annotationsfunktionen gibt es keine konsistente Benutzerschnittstelle für den Case Manager. Die Bedienung dieser Funktionen sowie deren Darstellung unterscheiden sich von Anwendung zu Anwendung. Um die Notwendigkeit des gedanklichen Umschaltens zwischen diesen Modalitäten für den Case Manager aufzuheben oder zumindest zu reduzieren, soll adCM ein Konzept für eine anwendungsübergreifend einheitliche Benutzerschnittstelle für die Anzeige und das Management von Annotationen enthalten.

4.2 KONZEPTIONELLE LÖSUNGSBAUSTEINE

Die adCM-Lösungsbausteine ordnen sie sich entlang der folgenden Verarbeitungskette ein. Sie ist – grob vereinfacht – am Vorgehen im Process Mining Lifecycle nach (van der Aalst et al. 2012) angelehnt und stellt in diesem Sinne eine spezielle Form des Process Mining für schwach strukturierte Prozesse dar (siehe Abbildung 10).

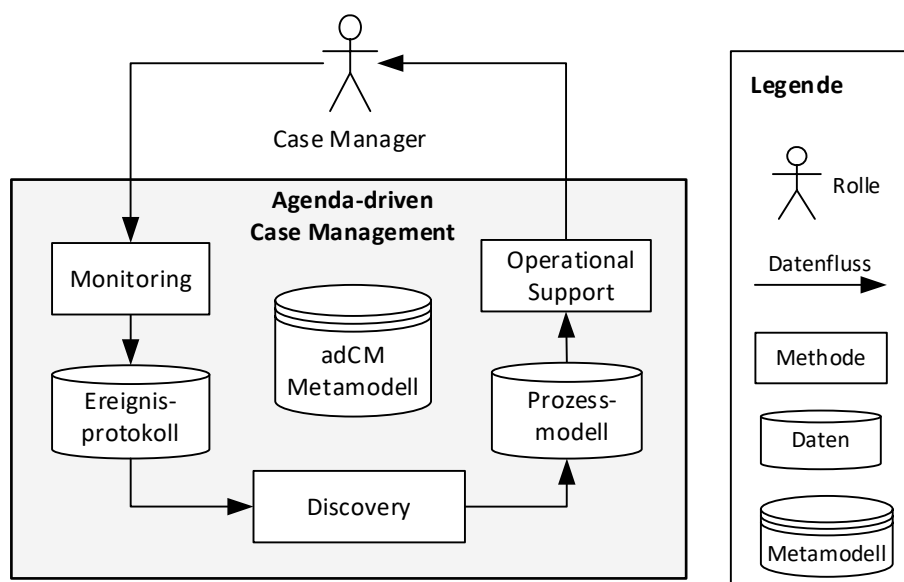


Abbildung 10: Verortung der adCM-Lösungsbausteine im Process Mining-Lifecycle

Die Verarbeitung beginnt mit dem Aufzeichnen von Case Management-Aktivitäten mithilfe von Ereignisprotokollen (**Monitoring**). Die Protokolle werden anschließend von Algorithmen analysiert, um Prozessmodelle zu gewinnen (**Discovery**). Im letzten Schritt werden die Modelle eingesetzt, um den Case Manager während der Prozessausführung zu unterstützen, beispielsweise durch Vorschläge für zukünftige Aktivitäten (**Operational Support**). Alle Methoden beziehen sich auf Konzepte, die im **adCM-Metamodell** definiert sind.

Das Metamodell, das Monitoring und das Discovery stellen die wesentlichen Lösungsbausteine des adCM dar. Sie werden daher in diesem Abschnitt kurz separat vorgestellt. Beim Operational Support handelt es sich um einen eher technischen Lösungsbaustein, der sich stark auf die Werkzeugunterstützung des Case Managers bezieht. Deshalb werden Details darüber, wie dem Case Manager zukünftige Aktivitäten konkret vorgeschlagen werden, im Rahmen der Beschreibung des Designs und der Implementierung des adCM-Werkzeugs in Kapitel 7.5 vorgestellt.

4.2.1 *ADCM-METAMODELL*

Das adCM-Metamodell stellt die konzeptionelle Grundlage dieser Arbeit dar, indem es die übergeordneten Konzepte definiert. Es handelt sich dabei um eine Weiterentwicklung der in (Benner et al. 2012) publizierten „Open Process Cloud Concepts“. Alle darauf aufbauenden Lösungsbausteine bedienen sich dieser Konzepte und beschreiben unterschiedliche Methoden, mit denen das Case Management unter Zuhilfenahme der Konzepte unterstützt werden kann.

Die wesentlichen Elemente des adCM-Metamodells sind die Agenda, der Workspace, das Artefakt und das Template. Eine Agenda ist ein Konzept, mit dem der Case Manager seinen Plan über die zukünftigen Aktivitäten strukturiert. In Anlehnung an eine digitale Fallakte dient der Workspace zur Strukturierung von Informationen, die zur Bearbeitung des Falls relevant sind. Um von den technischen Details zur Speicherung der unterschiedlichen Informationen zu abstrahieren, wird das Konzept der Artefakte eingeführt. Templates fassen Teile einer Agenda zusammen, die in vergangenen Fällen häufig gemeinsam verwendet wurden. Sie stellen statistisch verdichtete Informationen über den Prozess dar und ersetzen somit gewissermaßen das Prozessmodell, das im traditionellen Process Mining für die Bereitstellung des Operational Support vorgesehen ist.

4.2.2 *MONITORING*

Die adCM-Verarbeitungskette beginnt mit der Aufzeichnung von Ereignissen, die während des Geschäftsprozesses durch Interaktionen des Case Managers mit einem Informationssystem

erzeugt werden. Alle Konzepte, die sich dieser Aufgabe direkt oder indirekt zuordnen lassen, werden im Folgenden dem Lösungsbaustein „Monitoring“ zugeordnet.

Das Aufzeichnen der Ereignisse dient dem Ziel, die Informationen über die Prozessausführung zu sammeln. Unter Berücksichtigung der Zielstellung werden dabei hauptsächlich Ereignisse im Zusammenhang mit Planungsaktivitäten des Case Managers protokolliert. Zusätzlich werden aber auch Aktivitäten aufgezeichnet, die mit der Arbeit an der Fallakte verbunden sind, wie zum Beispiel das Recherchieren, Archivieren, Strukturieren oder Reflektieren von Daten.

Zu diesem Zweck wird ein Ereignisdatenmodell entworfen, das alle Tätigkeiten eines Wissensarbeiters abbilden kann. Im Gegensatz zum traditionellen Process Mining wird außerdem ein Konzept vorgestellt, nach dem die Ereignisse zentral und auf einer Abstraktionsebene protokolliert werden, die keine zusätzlichen Vorverarbeitungsschritte erfordern. Die besondere Herausforderung dabei ist, eine gute Balance für den Informationsgehalt zu finden. Denn werden die Benutzerinteraktionen zu detailliert aufgezeichnet, besteht die Gefahr, dass keine Gemeinsamkeiten zwischen vergangenen Fällen erkannt werden können. Wird umgekehrt zu grob protokolliert, können die Ergebnisse für deren konkreten Einsatz nicht spezifisch genug sein. Davon abgesehen ist das entstehende Ereignisprotokoll standardkonform strukturiert, sodass es mit Werkzeugen aus dem Process Mining verarbeitet werden kann.

Eine Methode zur Analyse der Komplexität von Ereignisprotokollen stellt die Schnittstelle zum Process Mining dar. Die dabei errechneten Metriken dienen als Indikator, ob der in dem Protokoll abgebildete Prozess schwach strukturiert ist (Benner-Wickner et al. 2014a). In Abhängigkeit vom Ergebnis wird die Anwendung des traditionellen Process Mining oder des adCM empfohlen. Das Monitoring sowie die Komplexitätsmetriken werden aufgrund ihrer thematischen Nähe gemeinsam in Kapitel 5 vorgestellt.

4.2.3 *DISCOVERY*

Die Informationen aus dem Ereignisprotokoll werden im Discovery-Schritt dazu verwendet, Muster in den vergangenen Fällen zu erkennen und für die spätere Verwendung modellhaft festzuhalten. Im traditionellen Process Mining kommen hierbei flussorientierte Prozessmodelle zum Einsatz, die nach Möglichkeit alle Pfade des Prozesses abbilden. Weil solche Modelle für schwach strukturierte Prozesse nicht geeignet sind, wird im adCM-Discovery ein alternatives Datenmodell vorgestellt. Es basiert auf dem adCM-Metamodell und erweitert es gemäß der Zielstellung so, dass es Muster in den Planungsaktivitäten beschreiben kann. Zudem wird darauf eingegangen, wie das Datenmodell mithilfe von Technologien des Semantic Web so ausgestaltet

werden kann, dass es um domänenspezifische Konzepte ergänzt werden kann (Benner-Wickner et al. 2015b).

Um diese Muster zu erkennen, enthält der Discovery-Lösungsbaustein einen Mining-Algorithmus. Er analysiert die Planungsaktivitäten der vergangenen Fälle und sucht nach häufig auftretenden Elementen in den Plänen. An dieser Stelle kommt zum Tragen, dass adCM die im Case Based Planning eingeführte hierarchische Dekomposition von Eigenschaften eines Plans adaptiert. Denn diese Hierarchien werden als Bäume interpretiert und mithilfe eines Tree Mining-Algorithmus auf häufig wiederkehrende Teilbäume untersucht.

Das Vorschlagen von Aktivitäten auf Grundlage dieser Muster ist Teil des „Operational Support“-Bausteins. Er stellt das letzte Element in der Verarbeitungskette dar und schließt den adCM-Zyklus. Neben dem Vorschlagen von Templates sind in diesem Baustein alle Funktionen zusammengefasst, die den Case Manager unmittelbar bei der Bearbeitung eines Falls unterstützen. Dazu zählt allen voran das Design einer Vorschlagsfunktion (Recommender System), die zu dem aktuellen Fallkontext passende Muster identifiziert und dem Case Manager vorschlägt. Dazu zählen aber auch die Konzepte zur Unterstützung der anderen Kernaktivitäten wie dem Recherchieren, Archivieren, Strukturieren und Reflektieren.

4.3 ADCM-METAMODELL

In diesem Kapitel werden in Anlehnung an (Benner et al. 2012) die grundlegenden Konzepte des adCM definiert. Es beginnt mit einer Strukturbeschreibung anhand eines Objektmodells, das im weiteren Verlauf als adCM-Metamodell bezeichnet wird. Die darin definierten Konzepte werden von den adCM-Lösungsbausteinen zur Unterstützung des Case Managements verwendet (siehe Abbildung 11). Zum Schluss werden die Konzepte auch aus einer dynamischen Sicht betrachtet. Sie zeigt, wie die adCM-Konzepte miteinander in Wechselwirkung treten. Dabei wird im Detail entlang eines durchgängigen Beispiel-Szenarios erklärt, wie die Kernaktivitäten aus Kapitel 2 mithilfe der Konzepte unterstützt werden sollen.

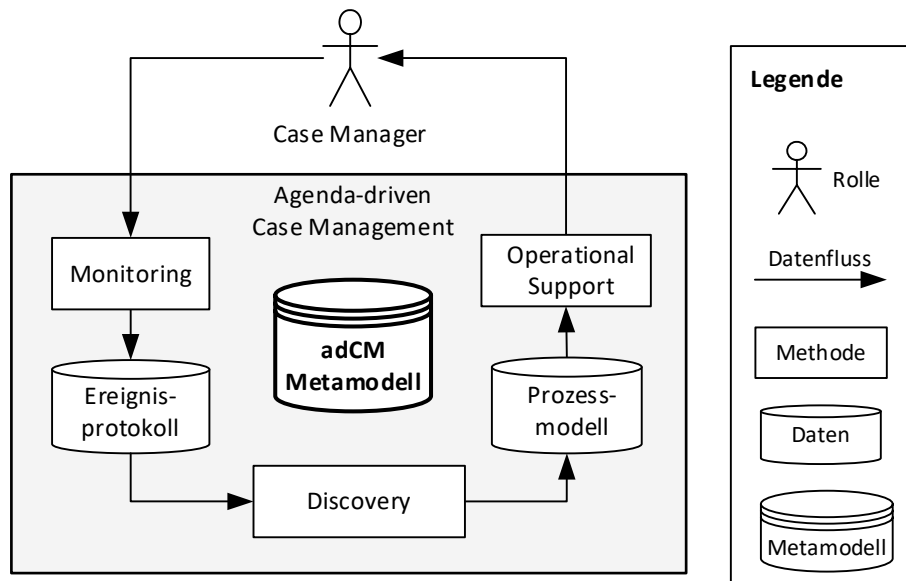


Abbildung 11: Lösungsbaustein „adCM-Metamodell“ im Gesamtkontext

4.3.1 AGENDA

Das grundlegende Element des Agenda-driven Case Managements ist die **Agenda**. Dabei handelt es sich um eine hierarchisch strukturierte Liste, die alle Elemente enthält, welche der Case Manager für die Ausführung eines Falls für wichtig erachtet. Die Elemente dienen dabei als Orientierungshilfe für die Planung des Falls. Der Case Manager ist nicht verpflichtet, diesen Plan exakt zu befolgen. In diesem Sinne repräsentiert die Agenda das zentrale Werkzeug, mit dem der Case Manager bei seinen **Planungstätigkeiten** unterstützt wird (vgl. Kapitel 2.2.2.1).

Im Idealfall ist die Agenda von einer Vorlage (Template) abgeleitet, damit auf bereits vorhandenem Erfahrungswissen über den Prozess (Prozesswissen) aufgebaut werden kann. Details zu den Templates folgen in Kapitel 4.3.3. Unabhängig von der Verwendung eines Template hat der Case Manager alle Freiheiten zur Strukturierung der Agenda. Er kann sie dem Fall und seinen besonderen Gegebenheiten frei anpassen oder sogar mit einer gänzlich neuen, leeren Agenda starten.

Um verschiedene Abstraktionsebenen zu unterstützen, kann eine Agenda hierarchisch organisiert werden. Dies gibt dem Case Manager die Möglichkeit, den Überblick über die wichtigsten Dinge und deren Zusammenhänge zu behalten. Dieser Ansatz steht in direktem Kontrast zum traditionellen Workflow Management, das die Art und Reihenfolge der Prozessausführung eng an vordefinierte Kontrollflüsse bindet.

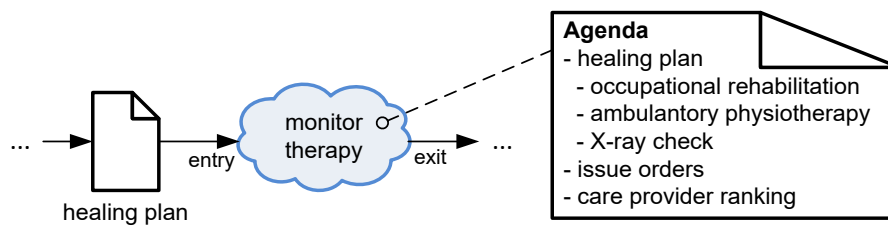


Abbildung 12: Beispiel-Agenda aus dem Rehabilitationsmanagement

Der Zweck und die Funktionsweise einer Agenda kann am besten anhand eines kurzen Beispiels aus dem Reha-Management erläutert werden (siehe Abbildung 12): Nachdem in einem vorausgegangenen Teilprozess ein Heilverlaufsplan für einen Patienten erstellt wurde, ist es nun die Aufgabe des Case Managers, die im Plan festgehaltenen Therapiemaßnahmen zu beobachten und bei Bedarf steuernd einzugreifen. Im Gegensatz zum vorangegangenen Prozessabschnitt handelt es sich nun um einen eher schwach strukturierten Abschnitt, dessen Aktivitätenreihenfolge nicht klar definiert werden kann (dargestellt durch eine Wolke). In einem solchen Abschnitt wird der Case Manager durch das Konzept der Agenda unterstützt. Sie könnte mit wegweisenden Begriffen wie „Heilverlaufsplan“, „Gutachten“ und „Leistungsträger“ ausgefüllt sein. Sie stammen von einem Template, das übliche Elemente einer Agenda des Typs „Therapiemaßnahme beobachten“ enthält. In diesem Beispiel sind die den genannten Einträgen untergeordneten Begriffe auf der Abstraktionsebene so konkret, dass sie bereits fallspezifisch sind und vom Case Manager selbst ergänzt wurden. Unter dem Element „Heilverlaufsplan“ hat er die einzelnen zu der Diagnose passenden Therapieschritte aufgeführt, wie zum Beispiel „Reha“, „ambulante Physiotherapie“ und „Röntgen-Check“.

Im adCM-Metamodell ist die Agenda Bestandteil eines Falls (Case). Umgekehrt ist dem Fall genau eine Agenda zugeordnet (siehe Abbildung 13). Die Agenda besteht aus einem Wurzel-Agendaeintrag, der mithilfe einer Eltern-Kind-Beziehung beliebig viele hierarchisch geschachtelte Einträge enthalten kann. Dadurch entsteht eine Baumstruktur. An dieser Stelle sei darauf hingewiesen, dass es sich bei dem Metamodell nicht um eine softwaretechnische, sondern eine fachliche Architektur handelt, weshalb die Baumstruktur nicht mit einem ausführlicheren Entwurfsmuster wie dem Kompositum (Gamma 1995) beschrieben wird.

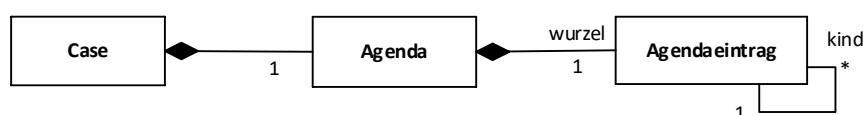


Abbildung 13: Auszug aus dem adCM-Metamodell: Aufbau der Agenda (UML-Klassendiagramm)

Weil die Agenda die Planungsaktivitäten eines Case Managers unterstützen soll, ist sie gemäß der Zielstellung das wichtigste Instrument im adCM. Deshalb orientieren sich um sie herum alle informationstechnischen Unterstützungsfunktionen. Die Agenda spiegelt mit dem darin enthaltenen Plan am ehesten die *kontrollflussorientierte* Perspektive des Prozesses wider. Inwiefern die Arbeit mit *Daten* konzeptuell unterstützt werden kann, wird im nachfolgenden Kapitel beschrieben.

4.3.2 WORKSPACE MIT ARTEFAKTEN

Neben dem Planen der zukünftigen Handlungsschritte sind nach (North und Guldenberg 2008) auch das **Recherchieren**, **Archivieren**, **Strukturieren** und **Reflektieren** von Informationen Aktivitäten, die in der Wissensarbeit anfallen. Insbesondere im Case Management – wie in Kapitel 2.3 dargelegt – wird diesen Aktivitäten im Zusammenhang mit der zentralen Fallakte eine wichtige Rolle zuteil. Ausgehend von den beschafften Informationen entwickelt der Case Manager eine Vorstellung davon, wie er den Fall „lösen“ will. Die dabei benötigten Datenquellen sind aber nach (Loeffeler et al. 1998) im Vorfeld nicht abschließend definierbar. Sie können sowohl aus dem Unternehmen stammen als auch von externen Dienstleistern oder Organisationen. Sie können strukturiert (Masken eines Informationssystems, Formulare, Formblätter) oder unstrukturiert (E-Mails, Notizen) vorliegen. Sie werden im weiteren Verlauf unter dem Begriff „**Artefakte**“ subsumiert. Das adCM-Konzept sieht vor, dass der Case Manager mit diesen Artefakten arbeiten kann, indem er die darin enthaltenen Informationen kombiniert, studiert, bewertet und in individuell zusammengestellte Arbeitsbereiche (**Workspaces**) einordnet (siehe Abbildung 14).

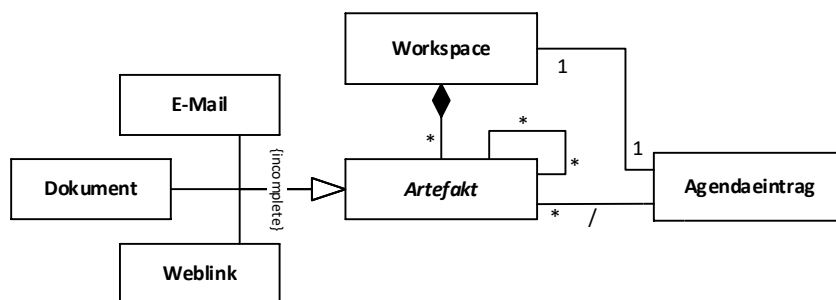


Abbildung 14: Auszug aus dem adCM-Metamodell: Aufbau des Workspace (UML-Klassendiagramm)

Die Vielgestaltigkeit der Artefakte ist in dem Metamodell durch eine unvollständige Menge an Unterklassen beschrieben – dargestellt durch die Eigenschaft „{incomplete}“. Artefakte selbst sind nicht instantiierbar, sondern existieren immer nur in Form einer konkreten Ausprägung beziehungsweise Unterklasse. Dieser Zusammenhang ist durch die abstrakte Klasse *Artefakt* beschrieben. Aus Sicht des Workspace ist aber der Typ eines Artefakts nicht relevant. Der Workspace dient hier als Container für Artefakte unabhängig ihres Typs.

Um die Strukturierung des Falls zu unterstützen, sollen die Artefakte einem bestimmten Teil des Plans zugeordnet werden können. Hierzu sieht das adCM-Metamodell vor, dass es zu jedem Agendaeintrag genau einen Workspace gibt, in dem alle dazugehörigen Artefakte abgelegt werden können. Aus dieser 1:1-Zuordnung lässt sich indirekt die Zuordnung zwischen einem Artefakt und einem Agendaeintrag ableiten (in UML dargestellt durch eine abgeleitete Assoziation: „/“). An dieser Stelle sei darauf hingewiesen, dass es sich bei diesem Agendaeintrag einerseits um einen sehr konkreten Teil des Plans handeln kann, der sehr tief in der Agenda-Hierarchie eingeordnet ist. Es kann sich aber auch um das Wurzelement der Agenda handeln, dessen Workspace fallübergreifend ist. Falls also die Zuordnung eines Artefakts zu einem Workspace für den Case Manager schwierig sein sollte, weil es sich etwa um ein zentrales, während des gesamten Falls relevantes Artefakt handelt, so kann es auch dem Workspace des Wurzeintrags zugeordnet werden. Das konkrete Design dieses Konzepts wird in 7.6.3.2 beschrieben.

Um Informationen zu **strukturieren**, können Artefakte nicht nur einem Agendaeintrag, sondern auch miteinander in Beziehungen gesetzt werden. Das wird im Metamodell über die reflexive, n-äre Assoziation zwischen Artefakten beschrieben. Im Gegensatz zu der reflexiven Beziehung in der Agendastruktur (siehe Abbildung 13), ist dieser Assoziation aber keine einschränkende Semantik (Eltern-Kind) mit entsprechenden Multiplizitäten zugeordnet. Sie können demnach auch einen Graphen bilden. Überlegungen dazu, wie diese Strukturen technisch realisiert werden können, sind Gegenstand des Kapitels 7.6.3.1.

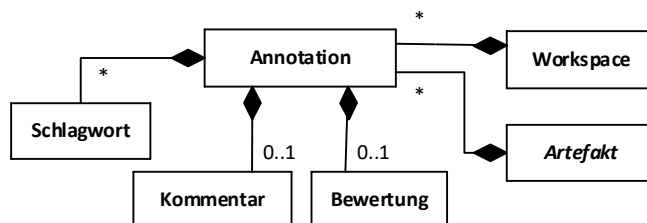


Abbildung 15: Auszug aus dem adCM-Metamodell: Annotationen (UML-Klassendiagramm)

Auch das **Reflektieren** von Informationen wird im Metamodell berücksichtigt. Hierzu wird das bereits in Kapitel 2.2.2.4 kurz eingeführte Konzept der Annotation verwendet (siehe Abbildung 15). Eine Annotation kann aus beliebig vielen Schlagwörtern, einem Kommentar oder einer Bewertung bestehen. Sowohl einzelne Artefakte als auch ein ganzer Workspace kann mit beliebig vielen Annotationen versehen werden. Auf dieser Abstraktionsebene wird noch nicht zwischen solchen Annotationen unterschieden, die sich auf ein Artefakt insgesamt beziehen und solchen, die sich auf Teile des Inhalts in dem Artefakt beziehen. Eine entsprechende Konkretisierung erfolgt zusammen mit weiteren Design-Entscheidungen im Kapitel 7.6.4.

Bezugnehmend auf das Beispiel aus dem Reha-Management handelt es sich bei dem abgestimmten Heilverlaufsplan um eines der wesentlichen Artefakte, das bei der Beobachtung der Therapie eine wichtige Rolle spielt. Es ist einem eigenen Agendaeintrag zugeordnet. Dieses Artefakt könnte generell übliche Therapieschritte enthalten oder auch davon abweichen, falls dies die konkrete Diagnose mit allen damit verbundenen Risiken und Begleitumständen erfordert. In einem solchen Fall kann als Begründung für diese Abweichung beispielsweise ein Kommentar oder ein wissenschaftlicher Aufsatz mit einer Beschreibung des alternativen Ansatzes vom Case Manager mit dem Heilverlaufsplan verknüpft werden.

Die Integration dieser flexiblen und vielfältigen Funktionen stellt eine besondere technische Herausforderung dar, auf die in dem Kapitel 7.6 detaillierter eingegangen wird.

4.3.3 *TEMPLATE*

Der Case Manager kann Fälle mit einer leeren Agenda starten und dann mit einer eigenen Agenda einen ganz individuellen Weg finden, den Fall zu bearbeiten. Doch es gibt auch Fälle, bei denen es sich anbietet, auf bereits vorhandene Erfahrungen aus dem Prozess (Prozesswissen) aufzusetzen. Solches Prozesswissen kann zum Beispiel häufige Aktivitäten in Form einer vorbereiteten Agenda darstellen. Es kann sich aber ebenso um nützliche Dokumentvorlagen in Form von Artefakten handeln. Zum Beispiel könnte ein Case Manager zu Beginn der Therapiesteuerung im Rahmen der initialen Strukturierung seiner Agenda und der Workspaces daran interessiert sein, welche Aktivitäten und Informationen andere Case Manager kürzlich bei einer ähnlichen Diagnose bearbeitet haben. Welche Experten kommen für eine Konsultation infrage? Welche medizinischen Kennzahlen müssen überwacht werden? Um diese Fragen zu beantworten, wird der Case Manager durch mit Kollegen geteiltes Prozesswissen unterstützt.

Für strukturierte Geschäftsprozesse ist dieses Prozesswissen üblicherweise wohlbekannt. Es wird manuell bei der Business Analyse oder automatisch beim Process Mining erhoben und in Form von Prozessmodellen festgehalten. Bei schwach strukturierten Prozessen hingegen liegen solche Modelle aufgrund der wenigen Gemeinsamkeiten zwischen den Fällen oft nicht vor und falls doch, dann auf einer sehr allgemein gehaltenen Abstraktionsebene mit nur wenigen generischen Schritten. Ansätze, ein solches Prozessmodell dennoch zu erheben, resultieren in schwer les- und auswertbaren „Spaghetti“-Modellen (van der Aalst 2011, S. 301–317). In Kapitel 6.2.1 wird detailliert auf solche Versuche und die entsprechenden Ergebnisse eingegangen.

Abgesehen von wenigen alternativen Ansätzen (Song und van der Aalst 2008) werden die Prozesse mit kontroll- beziehungsweise datenflussorientierte Notationen modelliert. Solche

Notationen sind mit dem Ziel entworfen worden, eine strikte Reihenfolge für die Aktivitäten vorzugeben. Aufgrund der Unplanbarkeit des Lösungswegs beziehungsweise der dafür benötigten Daten sind solche Notationen aber nicht zur Modellierung schwach strukturierter Prozesse geeignet.

Das in dieser Dissertation entwickelte Konzept wählt einen anderen Ansatz, um Prozesswissen für schwach strukturierte Prozesse zu repräsentieren. Es sieht dabei einen grundsätzlichen Paradigmenwechsel bei der Definition von Prozessmodellen vor: Anstatt dem Versuch nachzugehen, einen einheitlichen, übergreifend für alle Fälle detailliert spezifizierten Ablauf zu finden, sollen vielmehr unabhängige Prozesswissen-Bausteine bereitgestellt werden, die losgelöst voneinander auf unterschiedlichen Abstraktionsebenen definiert und passend zum Kontext beliebig zu einer fallspezifischen Lösung kombiniert werden können. Diese Bausteine werden unter dem Begriff „**Templates**“ eingeführt und beziehen sich – ähnlich einer Agenda – auf einen Baum aus Agendaeinträgen (siehe Abbildung 16).

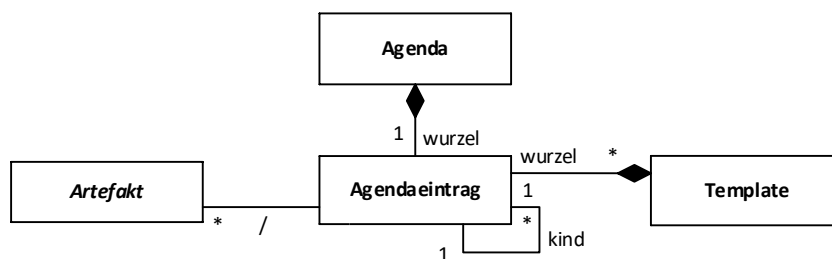


Abbildung 16: Auszug aus dem adCM-Metamodell: Templates (UML-Klassendiagramm)

Inwiefern die Templates unterschiedliche Abstraktionsebenen darstellen können, zeigt das Beispiel aus dem Rehabilitationsmanagement: Wie der Abbildung 12 entnommen werden kann, handelt es sich bei den Agendaeinträgen der ersten Gliederungsebene um generische Elemente, die in jedem Fall zu beachten sind und als Template infrage kommen. Der dadurch entstehende Mehrwert für den Case Manager ist noch verhältnismäßig gering, weil solche generischen Elemente aufgrund ihrer Häufung bereits in vielen Fällen Bestandteil der eignen Erfahrung sein werden. Die darunter einsortierten Einträge hingegen sind sehr spezifisch für den konkreten Fall und sind daher nur für einen ähnlichen Kontext nützlich. Zwar ist es aufgrund der eingangs beschriebenen Charakteristika von Case Management-Prozessen unwahrscheinlich, dass derselbe Fall sich wiederholt, doch bezogen auf einen isoliert betrachteten Teil der Agenda können sich durchaus spezifische Zusammenhänge beziehungsweise Muster häufen. Bei einer hinreichenden Häufung ist auch die Definition eines entsprechenden Template relevant, das genau diesen spezifischen Aspekt des Prozesswissens festhält, ohne dabei einen Anspruch auf Allgemeingültigkeit für alle Fälle zu haben. In diesem Sinne ist das Konzept unabhängiger,

kontextspezifischer Templates eine flexible Alternative zu einem übergreifenden Prozessmodell. Details zur Erzeugung und Beschreibung dieser Templates liefert das Kapitel 6.3.

Weil sich die Struktur der Templates aus der Struktur der Agenda ableitet, gelten für die Templates die gleichen Freiheitsgrade. Auf diese Weise können die Templates Best Practices eines Unternehmens, die persönlichen Erfahrungswerte eines Case Managers oder aggregierte Gemeinsamkeiten aus vergangenen Fällen widerspiegeln. An dieser Stelle sei angemerkt, dass das adCM-Konzept keinen Zwang zur Annahme eines passenden Templates durch den Case Manager vorsieht. Als Experte ist es seine individuelle Entscheidung, ob er verfügbare Templates ignoriert, Teile davon adaptiert oder sie als Ganzes in seine Agenda integriert. Die damit verbundenen Algorithmen und softwaretechnischen Design-Entscheidungen werden in Kapitel 7.5 beschrieben.

Im Gegensatz zu einem umfassenden, kontrollflussorientierten Prozessmodell können Templates dafür verwendet werden, einzelne, häufig durchgeführte Aktivitäten in die Agenda aufzunehmen, ohne dass sie in ein fest vorbestimmtes prozessuales Umfeld (zum Beispiel vorausgehende oder nachfolgende Aktivitäten) eingebettet werden müssen. In diesem Sinne handelt es sich bei den Templates um einen Ansatz zur deklarativen Beschreibung von Prozessen.

Weil Agendaeinträgen über deren Workspace Artefakte zugeordnet sein können, sieht das Konzept indirekt auch vor, Templates um häufig wiederkehrende Artefakte zu ergänzen. Am Beispiel des Reha-Managements ist der Heilverlaufsplan ein solches zentrales Dokument (beziehungsweise die entsprechende Dokumentenvorlage).

Templates können auch für die Definition von Rahmenbedingungen herangezogen werden, die bei Bedarf bei der Wiederholung ähnlicher Prozessinstanzen festgelegt werden können. Das Template kann dann etwa einen Zielzustand beschreiben, den ein Fall erreichen soll. Beispielsweise kann ein Prozessverantwortlicher die Regel festlegen, dass spätestens beim Abschließen eines Falls im Agendaeintrag für die letzten Case Management-Phase „Evaluation“ ein finales, ärztliches Gutachten vorliegen muss. Weiterführende Details zu der Integration von Geschäftsregeln in das adCM-Konzept werden in (Benner-Wickner et al. 2015b) beschrieben.

4.3.4 GESAMTSICHT UND FALLBEISPIEL

In diesem Abschnitt werden die zuvor einzeln beschriebenen Ausschnitte des adCM-Metamodells im Gesamtkontext sowie in einem beispielhaften Einsatz gezeigt. Hierzu werden drei Ebenen eingeführt: Die **Metamodell**- die **Modell**- und die **Instanzebene**. Dabei werden auch zusätzliche

Elemente beschrieben, die aufgrund ihrer untergeordneten Relevanz im adCM-Metamodell keine separate Beschreibung im Vorfeld benötigten. In Gestalt eines UML-Klassendiagramms (siehe Abbildung 17, oben) wird zuerst die **Metamodell-Ebene** dargestellt. Dabei handelt es sich um die Definition der Elemente, die in einem adCM-Modell verwendet werden können und wie sie miteinander in Beziehung stehen dürfen.

In der **Modell-Ebene** sind alle fallübergreifenden Instanzen des Metamodells enthalten. In dieser Ebene ist mit den Templates und den abgeschlossenen Fällen das gesammelte Prozesswissen beschrieben. Es wird manuell von den Prozessverantwortlichen gepflegt und kann mithilfe von Discovery-Algorithmen (siehe Kapitel 6) automatisch ergänzt werden. In der Mitte der Abbildung 17 werden beispielhaft zwei Templates gezeigt, die zu dem Reha-Management gehören.

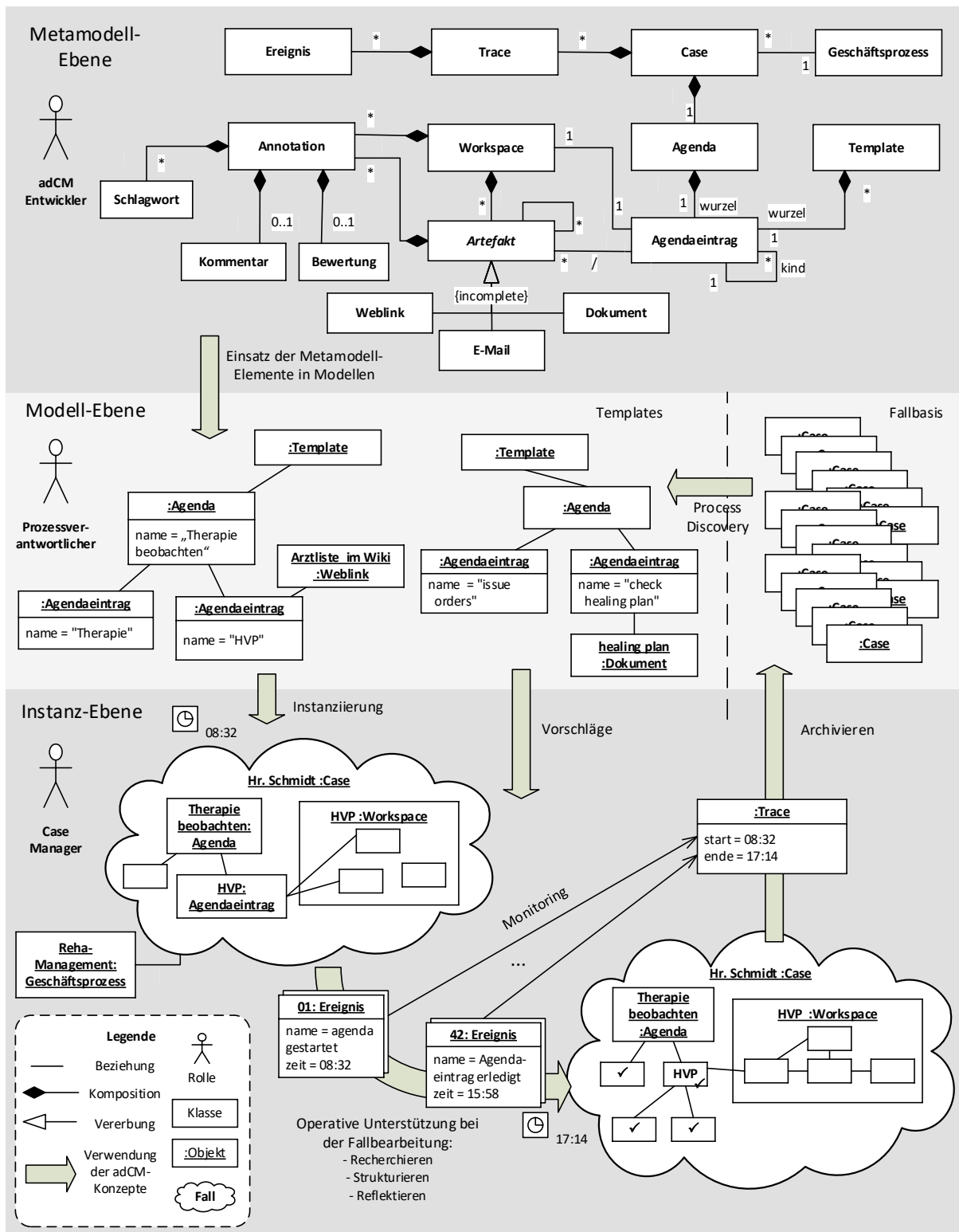


Abbildung 17: Definition des adCM-Metamodells und Einsatz der darin enthaltenen Konzepte auf Modell- und Instanzebene

Das Zusammenspiel der adCM-Konzepte aus **Modell- und Instanzebene** beschreibt einen Zyklus, der dem Case Based Reasoning ähnelt (siehe Abbildung 17, unten). Er wird bei jedem neuen Fall

wiederholt und führt zu Zustandsveränderungen in Ausprägungen (Instanzen) der adCM-Konzepte. Die einzelnen Schritte werden im Folgenden an einem Fallbeispiel beschrieben.

Herr Müller, Mitarbeiter im Reha-Management, ist verantwortlich für die Steuerung der Reha von Herrn Schmidt. Er legt morgens um 08:32 Uhr einen neuen Fall für die Beobachtung der Therapie an und steht vor der Entscheidung, eine gänzlich neue Agenda mit einem leeren Workspace zu starten, oder mit einer teilausgefüllten Agenda zu beginnen, die aus einem Template stammt. Er entscheidet sich für Letzteres und startet die Bearbeitung. Diese Aktion löst ein Ereignis aus, das in einem Trace festgehalten wird. Ein Trace bündelt eine Menge von zusammenhängenden Ereignissen, die während der Bearbeitung eines Falls auftreten (ein detailliertes Konzept zur Modellierung der Ereignisdaten folgt in Kapitel 5.3.1).

Mit der Fallbearbeitung beginnt ein vorher nicht determinierter Prozess, in dem Herr Müller den aktuellen Stand der Therapiemaßnahmen nachvollzieht, mit den im Heilverlaufsplan (HVP) festgehaltenen Zielen abgleicht und bei Bedarf gegensteuert oder die Ziele in Abstimmung mit Herrn Schmidt und den behandelnden Ärzten neu ausrichtet. Hierzu recherchiert er moderne Therapiemethoden, holt Meinungen von Gutachtern ein und setzt sie mit den Entscheidungen der Ärzte in Bezug und reflektiert den Erfolg einzelner Therapieschritte. Um einen Überblick über die Aktivitäten und ihren Bearbeitungsstand zu behalten, ordnet er die Tätigkeiten in seiner Agenda ein und kennzeichnet abgeschlossene Einträge.

Während einer Therapiemaßnahme tritt plötzlich ein psychisches Krankheitsbild bei Herrn Schmidt auf. Um diese neue Situation zu dokumentieren, fügt Herr Müller einen neuen Agendaeintrag hinzu, auf dessen Workspace er die Schilderung des Therapeuten ablegt. In diesem Moment erkennt das System eine teilweise Übereinstimmung der Agenda mit einem Template. Dabei handelte es sich um eine Aufzählung ergänzender Therapiemaßnahmen zur Behandlung psychischer Unfallfolgen mit entsprechenden Links zu den Vertragskliniken. Herrn Müller wurde das Template angeboten und er hat sich einen Teil der Therapiemaßnahmen, die für den Heilverlauf von Herrn Schmidt infrage kommen, in die Agenda eintragen lassen.

Einige Wochen später ist die Therapie abgeschlossen. Herr Müller sortiert kurz vor Dienstende um 17:14 das Protokoll des Abschlussgesprächs zusammen mit dem Evaluationsbericht in die Fallakte ein, markiert den Fall als abgeschlossen und archiviert ihn in die Fallbasis.

Dieser Zyklus beschreibt aufbauend auf das Metamodell die dynamischen Aspekte des adCM. Die Algorithmen, konkreten Datenstrukturen und Technologien, die für die Realisierung eines solchen Zyklus erforderlich sind, werden jeweils in den Lösungsbausteinen beschrieben.

4.4 ZUSAMMENFASSUNG

In diesem Kapitel wurden die zentralen Konzepte des Agenda-driven Case Management vorgestellt:

- Die **Agenda**: Eine flexible, unverbindliche, hierarchisch strukturierte Liste zur Organisation aller Dinge, die ein Case Manager zur Steuerung des Falls für wichtig erachtet. Sie ist das zentrale Konzept zur Unterstützung von **Planungstätigkeiten**.
- Der **Workspace mit Artefakten**: frei gestaltbarer Arbeitsbereich, der im Sinne einer Fallakte eine strukturierte Ablage für die benötigten Informationen bietet. Zusammen mit den Annotationen bildet er für jeden Agendaeintrag das zentrale Konzept zur Unterstützung des **Recherchierens, Archivierens, Strukturierens** und **Reflektierens**.
- Die **Templates**: Eine Menge unabhängiger, kontextabhängiger Bausteine aus hierarchisch verschachtelten Agendaeinträgen, die Erfahrungswissen über den Prozess beschreiben und im Falle eines passenden Kontexts wiederverwendet werden können. Als Gegenentwurf zum imperativen Prozessmodell stellen sie eine flexible Alternative zur Beschreibung von Prozesswissen dar und unterstützen die **Planungsaktivitäten** eines Case Managers.

Die Dynamik des adCM-Konzepts beschreibt einen Zyklus, in dem der Case Manager zu Beginn eines Falls mit einer leeren Agenda startet, oder mit einer Agenda, die bereits mit einem Template gefüllt wurde. Während der Bearbeitung des Falls werden Artefakte zusammengetragen, einsortiert und bewertet. Die Agenda wird flexibel an die jeweiligen Erfordernisse angepasst. Entsteht dadurch eine Situation, zu der ein Template existiert, wird dieses Template dem Case Manager zur Integration in die aktuelle Agenda vorgeschlagen. Zum Schluss wird der Fall in die Fallbasis geschrieben. Von dort aus können Discovery-Algorithmen die Fälle und deren Veränderungen entlang der Zeitachse untersuchen und anhand der entdeckten Muster neue Templates erzeugen.

Beim adCM-Metamodell handelt es sich um Konzepte, die jeweils einen bestimmten Grundgedanken des adCM aufgreifen und ausprägen. Sie bilden die konzeptionelle Grundlage für die im weiteren Verlauf der Dissertation vorgestellten Lösungsbausteine.

5 MONITORING

Das Monitoring im Process Mining befasst sich mit der Aufgabe, die Interaktionen des Benutzers mit den verwendeten Informationssystemen während der Ausführung des Geschäftsprozesses zu protokollieren (siehe Abbildung 18). Diese Protokolle (Event-Logs) spielen in zweierlei Hinsicht eine wichtige Rolle in adCM. Erstens bauen alle weiteren adCM-Lösungsbausteine auf die Protokolle des Monitorings auf. Zweitens kann die Komplexität der Protokolle Aufschluss darüber geben, ob es sich entweder um einen eher strukturierten Prozess handelt, der mit den bekannten Discovery-Methoden des Process Mining analysiert werden kann, oder um einen eher schwach strukturierten Prozess, der durch die gezielte Anwendung von Methoden des Agenda-driven Case Management unterstützt werden sollte.

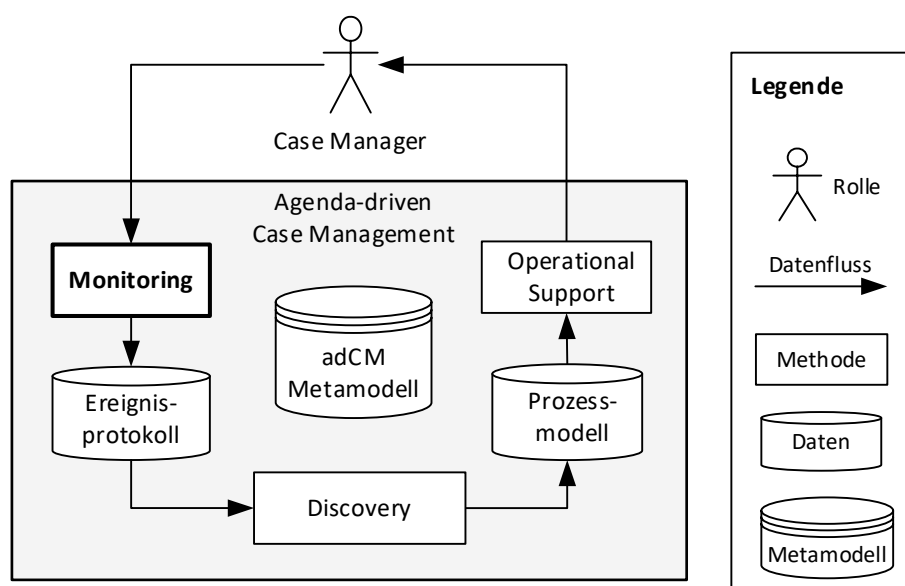


Abbildung 18: Lösungsbaustein "Monitoring" im Gesamtkontext

In weiteren Verlauf dieses Kapitels werden zu Beginn die Herausforderungen benannt, die im Zusammenhang mit dem Monitoring auftreten und welche konkreten Anforderungen sich daraus ableiten lassen. Daraufhin werden verwandte Ansätze diskutiert und deren offene Punkte hervorgehoben. Im Lösungskonzept werden diese Lücken und Anforderungen aufgegriffen und mit verschiedenen Modellen und Methoden adressiert. Zum Schluss werden die einzelnen

Elemente des Lösungskonzepts evaluiert und zusammengefasst. Details zur Implementierung des Lösungsbausteins finden sich in Kapitel 7.3.

5.1 PROBLEMSTELLUNG

Zu Beginn dieses Kapitels wird auf die Herausforderungen eingegangen, die im Zusammenhang mit dem Monitoring von Geschäftsprozessen auftreten. Als Grundlage dienen dabei diejenigen Herausforderungen, die in (van der Aalst et al. 2012) hinsichtlich des Monitorings **strukturierter** Geschäftsprozesse identifiziert wurden. Dabei wird erstens diskutiert, weshalb diese Herausforderungen auch für schwach strukturierte Prozesse gelten und zweitens inwiefern sie sich aufgrund der Besonderheiten schwach strukturierter Geschäftsprozesse sogar verschärfen. Im Fokus dieser Diskussion liegen die Anforderungen an die Qualität von Ereignisdaten und die Berücksichtigung einer Vielzahl unterschiedlicher Informationssysteme. Die durch das Monitoring zu protokollierenden Aktivitäten ergeben sich aus dem adCM Metamodell: Das Arbeiten mit der Agenda und den Artefakten. Zusätzlich soll das adCM Monitoring die Aufgabe übernehmen, den Kontext eines Falls als wertvollen Input für nachfolgende Schritte der adCM-Methode zu identifizieren.

5.1.1 HERAUSFORDERUNGEN BEIM MONITORING SCHWACH STRUKTURIERTER PROZESSE

In der gesamten Verarbeitungskette spielt die **Qualität der Ereignisdaten** eine herausragende Rolle, denn auf diese Daten bauen alle Process Mining-Methoden auf. Um die geforderte Qualität zu erreichen, ist zu Beginn eines Process Mining-Projekts eine zusätzliche Projektphase vorgesehen. Sie umfasst die manuelle Identifikation der Ereignisprotokolle in den betroffenen Informationssystemen, diese hinsichtlich des Erkenntnisinteresses zu filtern und in ein gemeinsames Zielprotokoll zu übertragen. Doch nicht alle Informationssysteme protokollieren Ereignisse und selbst wenn, liegen die Protokolle nicht ohne weiteres Zutun in einer standardisierten Form vor. Denn bislang sind solche Daten üblicherweise nur Nebenprodukte in Informationssystemen, zum Beispiel für die Fehleranalyse. Für die Ausführung von Discovery-Algorithmen wird aber eine einzige Ereignisdatenquelle mit einer klar definierten Struktur und Semantik benötigt (van der Aalst et al. 2012).

Bezogen auf das Monitoring von schwach strukturierten Prozessen spitzt sich diese Herausforderung zu. Die Ursache hierfür liegt in einer Kette von Zusammenhängen: Zunächst handelt es sich bei schwach strukturierten Geschäftsprozessen definitionsgemäß um Prozesse, in denen der Lösungsweg oder die benötigten Datenquellen schwer vorhersehbar sind. Hinzu

kommt nun, dass je nach Art der Datenquelle oft ein spezielles Informationssystem eingesetzt wird, um diese Datenquelle zu erschließen beziehungsweise zugreifbar zu machen. Beide Sachverhalte zusammengekommen bewirken, dass eine abschließende Identifikation aller für das Process Mining relevanter Informationssysteme schwierig ist. Will man also sicherstellen, dass zu allen benötigten Datenquellen auch Ereignisprotokolle existieren, müssten demnach **alle** Anwendungen und Informationssysteme berücksichtigt werden. Kann eine Organisation diese Aufgabe nicht leisten (zum Beispiel wegen mangelnder Ressourcen), sorgt dieser Sachverhalt für eine Reduktion der Datenbasis und damit für eine Verschlechterung der Ereignisdaten. Deshalb müssen beim Mining schwach strukturierter Geschäftsprozesse **mehr Informationssysteme** analysiert werden als bei strukturierten Geschäftsprozessen.

Ein weitere Verschärfung des Problems im Kontext schwach strukturierter Prozesse ist der **Bedarf nach semantisch verknüpften Ereignisdaten**, eine der vier Qualitätskriterien aus (van der Aalst et al. 2012). Denn selbst wenn es gelingt, alle relevanten Informationen über den Prozessverlauf in einem gemeinsamen Ereignisprotokoll zusammenzuführen, stellt sich das Problem, Zusammenhänge in diesen unterschiedlichen Informationen zu identifizieren und den Zielen des Geschäftsprozesses zuzuordnen.

Ein Beispiel-Szenario soll zeigen, wie sich diese Herausforderungen im Kontext des Reha-Managements eines gesetzlichen Unfallversicherers darstellen:

Die Poststelle einer gesetzlichen Unfallversicherung erhält den Bericht eines Durchgangsarztes. In dem Schreiben geht es um die Erstversorgung eines während der Arbeit verunfallten Patienten, Herrn Schmidt. Die Poststelle scannt das Schreiben ein und erzeugt dadurch einen neuen Fall mithilfe eines mit einem Informationssystem zur Fallverwaltung verknüpften Dokumentenmanagementsystems. Ein zugewiesener Reha-Manager sieht den neuen Fall und beginnt mit der Bearbeitung, indem er das Schreiben liest und sich Notizen in einem Word-Dokument macht. Falls vorhanden, gibt er in der Fallverwaltung den Diagnoseschlüssel ein, anhand dessen eine für das Controlling wichtige Laufzeitprognose errechnet werden kann. Gibt es keinen solchen Schlüssel, sucht er nach einem geeigneten Arzt für eine manuell erstellte Prognose des Heilverlaufs. Hierzu greift er auf eine Liste mit bekannten Spezialisten zu, die er vor einigen Jahren in Excel angelegt hat und seitdem pflegt. Um Handlungsanweisungen für bestimmte Situationen zu erhalten, greift er über das Internet auf eine Seite des Dachverbandes zu. Weil der Patient in diesem Fall schwerverletzt ist, informiert sich der Reha-Manager mithilfe einer entsprechenden PDF-Datei über das Schwerstverletztenartenverfahren (SAV). Gelegentlich

muss er auch die gesetzlichen Rahmenbedingungen für seine Beratungstätigkeit nachschlagen, wie zum Beispiel das Recht auf Verletztengeld gemäß §§ 45–48 SGB VII.

Anhand dieses Szenarios kann man bereits erkennen, wie vielfältig und fallspezifisch die Wahl der Informationssysteme und Datenquellen ist. Ausgehend davon werden die folgenden Anforderungen an das adCM-Konzept gestellt.

5.1.2 ANFORDERUNGEN AN DAS ADCM-MONITORING

Anforderung 9: Events protokollieren

Die zentrale Aufgabe der Monitoring-Komponente ist es, alle relevanten Ereignisse zu protokollieren. Aufgrund der hohen Zahl unterschiedlicher Informationssysteme ist es aber schwierig, alle relevanten Ereignisse zu erhalten. **Das Monitoring-Lösungskonzept soll zur Klärung der folgenden Fragen beitragen:**

- **Welche Ereignisse im Zusammenhang mit der Abarbeitung der Agenda müssen protokolliert werden**, damit die nachfolgenden adCM-Lösungsbausteine Wissen über den Prozessablauf sammeln und bereitstellen können?
- **Welche Ereignisse müssen bei der Arbeit mit Artefakten aufgezeichnet werden**, um später zum Beispiel allgemein oder speziell für Agenda-Einträge benötigte Artefakte ermitteln zu können?
- **Welche zusätzlichen Ereignisse müssen protokolliert werden, um alle Kernaktivitäten der Wissensarbeit festhalten zu können?**

Anforderung 10: Qualität des Ereignisprotokolls

Die zu protokollierenden Events müssen eine bestimmte Qualität besitzen, damit die nachfolgenden Mining-Schritte erfolgreich sein können. Für das Process Mining werden nach (van der Aalst et al. 2012) fünf verschiedene Qualitätsstufen für Ereignisprotokolle definiert. Aufgrund der im Kapitel 5.1.1 motivierten besonderen Herausforderungen beim Monitoring schwach strukturierter Prozesse müssen diese Protokolle konsequenterweise eine höhere Stufe erreichen als Protokolle strukturierter Prozesse. Weil zuletzt genannte Protokolle nach (van der Aalst et al. 2012) mindestens die Stufe 3 erreichen sollten, sollte die Qualität der zuerst genannten Protokolle also die Stufe 4 oder 5 erreichen. Die besten Voraussetzungen für das Mining-Ergebnis erhält man demnach, wenn die Protokolle verlässlich und vollständig sind und einer wohldefinierten Struktur folgen. Hierzu sollten sie automatisch, systematisch, zuverlässig und sicher aufgezeichnet werden, unter Beachtung von Datenschutz- und Datensicherheitsbestimmungen.

Die Datenstruktur der Events sollte so beschaffen sein, dass die Attribute eine eindeutige und bekannte Semantik haben. Um das zu erreichen, sollten die verwendeten Begriffe aus einem zuvor abgestimmten Vokabular oder einer Ontologie stammen beziehungsweise darauf verweisen (van der Aalst et al. 2012).

Das Monitoring-Lösungskonzept soll eine verlässliche, vollständige, wohldefiniert strukturierte, automatische, systematische, zuverlässige und semantisch verknüpfte Protokollierung unterstützen.

Anforderung 11: Komplexitätsmetriken für das Ereignisprotokoll

Weil die Ereignisprotokolle eine essentielle Informationsquelle für die nachfolgenden Discovery-Algorithmen sind, ist ihre Qualität von zentraler Bedeutung für den Erfolg des Process Discovery. Beispielsweise sollten sie möglichst vollständig sein („Completeness“) und ausschließlich relevante Aktivitäten enthalten („Noise-Free“). Viele Studien haben sich in den letzten Jahren aber hauptsächlich auf die Qualität der erzeugten Prozessmodelle konzentriert oder inwiefern komplizierte, aus der Praxis stammende Protokolle so vorverarbeitet werden können, dass sie den Ansprüchen der Algorithmen genügen. Beispiele hierfür sind (Rozinat und van der Aalst 2008), (Greco et al. 2006) und (van der Aalst et al. 2006). Es wurden zudem spezielle Discovery-Algorithmen entwickelt, um mit unvollständigen und rauschbehafteten Protokollen besser umgehen zu können. Dazu zählen der Genetic Algorithm (de Medeiros et al. 2007), der Fuzzy Miner (Günther und van der Aalst 2007) und der Cluster Miner (de Medeiros et al. 2008). Bei diesen Ansätzen bleibt aber ein Gedanke unberücksichtigt: Die offenbar mangelhafte Vollständigkeit und der hohe Anteil an mutmaßlich irrelevanten Aktivitäten müssen nicht zwangsläufig schlechte Protokollqualität bedeuten. Vielmehr könnte diese augenscheinlich schlechte Qualität auch eine Auswirkung der Komplexität eines schwach strukturierten Prozesses sein. Die Anwendung traditioneller Discovery-Algorithmen auf schwach strukturierte Prozesse ist aber nicht besonders erfolgreich, wie später noch in Kapitel 6.4 gezeigt wird. Deshalb werden Metriken benötigt, mit denen im Vorfeld gemessen werden kann, ob die Qualität der Protokolldaten für die Anwendung von traditionellen Discovery-Algorithmen überhaupt ausreichend ist. Solche Forschungsfragen haben aber in der Process Mining-Community bislang kaum Beachtung gefunden. Insbesondere vor dem Hintergrund der wesentlichen Merkmale schwach strukturierter Prozesse gibt es keinen Grund, weshalb sich deren Komplexität nicht auch auf die Ereignisprotokolle auswirken sollte. Aufgrund dieser Komplexität ist zu erwarten, dass die Anwendung von Discovery-Algorithmen ohne vorgeschaltete Qualitätsprüfung nur schwer auswertbare und insbesondere nicht menschenlesbare „Spaghettiprozesse“ entstehen lässt. Um

das zu vermeiden, sollte vielmehr im Vorfeld gemessen werden, ob die Komplexität des Ereignisprotokolls überhaupt ausreicht. Diese Forderung begründet sich außerdem darin, dass Discovery-Algorithmen eine hohe Zeitkomplexität haben, die sich auf bis zu mehrere Wochen erstrecken kann (Verbeek und van der Aalst 2013).

Das Monitoring-Lösungskonzept sollte daher zum Inhalt haben, solche Komplexitätsmetriken für Ereignisprotokolle zu entwickeln. Sie sollen in erster Linie eine Art Indikator darstellen, ob die Qualität eines Ereignisprotokolls hinreichend für die Anwendung traditioneller Discovery-Algorithmen ist. Wird die Qualität als zu schlecht eingestuft, weisen sie zugleich auf die Anwendung agilerer Techniken hin, wie zum Beispiel dem Agenda-driven Case Management.

Anforderung 12: Kontext des Falls ermitteln

In Anlehnung an das von Aamodt und Plaza eingeführte Konzept des Case-based Reasoning (Aamodt und Plaza 1994), sollen die adCM-Konzepte dafür verwendet werden, Lösungen aus vergleichbaren Fällen vorzuschlagen, die in einer Wissensbasis vorgehalten werden. Das Adjektiv „vergleichbar“ spielt dabei eine wesentliche Rolle: Wie bereits im Kapitel 4.3.3 beschrieben, sind Templates nur in einem bestimmten Kontext nützlich. Um die Identifizierung eines zu einem Kontext passenden Templates zu vereinfachen, muss der Kontext bekannt sein. **Das Monitoring soll eine Lösung dafür finden, was als Kontextinformation eines Falls infrage kommen könnte und wie diese Kontextinformationen zur Laufzeit an die nachgelagerten Lösungsbausteine weitergereicht werden können.**

5.2 RELATED WORK

In diesem Abschnitt werden zunächst die verwandten Ansätze aus der Process Mining-Community benannt, die sich mit dem Monitoring beschäftigen. Anschließend wird kurz auf die Dissertation von Andreas S. Rath eingegangen, der ebenfalls ein Konzept zur Erfassung und semantischen Klassifizierung von Tasks am PC-Arbeitsplatz erarbeitet hat.

5.2.1 MONITORING IM PROCESS MINING WORKFLOW

Im Process Mining Workflow (van der Aalst 2011) wird ein Weg beschrieben, wie man von den unterschiedlichen Ereignisdatenquellen zu einer solchen zentralen Quelle gelangt. Dabei wird das Paradigma verfolgt, dass nicht alle verfügbaren Informationen unreflektiert als Quelle hinzugezogen werden, sondern dass vielmehr anhand vorab konkretisierter Fragestellungen

gezielt bestimmte Quellen ausgewählt werden. Diese Quellen werden dann per ETL¹-Methoden geladen oder liegen bereits in einem Data Warehouse-System (falls vorhanden) vor. Nicht immer handelt es sich bei den Quellen um Ereignis-Daten im eigentlichen Sinne, sondern zum Beispiel um Auftragslisten. Aus solchen Informationen müssen dann erst Ereignisdaten erstellt werden. Dazu zählt auch die Zuordnung von Ereignissen zu dem zu analysierenden Prozess; schließlich ist es sehr wahrscheinlich, dass die Daten auch Informationen enthalten, die im Kontext anderer Prozesse entstanden sind.

Obwohl der Process Mining Workflow bereits durch mächtige Werkzeuge wie das in Kapitel 2 genannte ProM-Framework unterstützt werden kann, handelt es sich um einen sehr schwergewichtigen Prozess mit arbeitsintensiven Analysephasen. Ziel der Dissertation ist es, einen leichtgewichtigeren Ansatz zu finden, um zentrale und standardisierte Ereignisdaten zu erzeugen.

5.2.2 MONITORING ZUR IDENTIFIKATION DER AKTUELLEN AUFGABE EINES WISSENSARBEITERS

Auch Andreas S. Rath beschäftigt sich in seiner Dissertation (Rath 2010) mit Möglichkeiten zur Aufzeichnung von Benutzerinteraktionen von Wissensarbeitern. Mit dem Ziel, die aktuelle Aufgabe eines Wissensarbeiters automatisch zu identifizieren, werden dessen Interaktionen protokolliert und mit einer speziell dafür entworfenen Ontologie über den Interaktionskontext („User Interaction Context Ontology“, UICO) verknüpft. Um die übergeordnete Aufgabe zu identifizieren, müssen die auf einer sehr technischen Ebene aufgezeichneten Events wie „Ausschneiden“ oder „Einfügen“ über mehrere Transformationsschritte zu höherwertigen Ereignissen zusammengeführt werden.

Die Konzepte Agenda, Workspace und Template erlauben hingegen einen pragmatischeren Weg, Aussagen über die Aktivitäten des Benutzers zu erhalten als die von Rath entwickelte UICO. Das liegt daran, dass die Protokollierung auf einer höheren Abstraktionsebene erfolgt: Weil die Agendaeinträge bereits eine Art Repräsentation der Aufgaben auf einer fachlichen Ebene darstellen, erlauben die Interaktionen mit den Agendaeinträgen bereits Rückschlüsse auf die aktuelle Aufgabe.

¹ Extract-Transform-Load

5.2.3 ANSÄTZE ZUR MESSUNG DER EREIGNISPROTOKOLL-KOMPLEXITÄT

Metriken zur Messung von Komplexität spielen eine wichtige Rolle bei der Analyse von Geschäftsprozessen. In den letzten Jahren wurden sowohl für Prozessmodelle als auch für Ereignisprotokolle entsprechende Metriken entwickelt. Beispielsweise haben (Günther und van der Aalst 2007) drei Typen von Prozessmetriken entwickelt, die auf der Grundlage von Ereignisprotokollen ermittelt werden: unäre und binäre Signifikanz sowie binäre Korrelation. Allerdings muss für die Berechnung dieser Metriken erst ein kompliziertes Modell aus paarweisen Beziehungen zwischen den Ereignissen erstellt werden. Das heißt, die Metriken basieren gar nicht unmittelbar auf den Ereignisprotokollen selbst.

Van der Aalst et al. haben schon 2009 eine Methode vorgestellt, wie man mithilfe von Aggregation und Abstraktion Modelle hinsichtlich ihrer Komplexität vereinfachen kann (van der Aalst 2009). Unter Abstraktion verstehen die Autoren in diesem Kontext das Entfernen weniger relevanter Ereignisse. Das ist im Kontext variantenreicher Prozesse aber problematisch, weil die Ereignisse in den Protokollen hinsichtlich ihrer Häufigkeit sehr breit gestreut sind. Diese Streuung sorgt für eine stete, aber nur selten sprunghafte Abnahme an Häufigkeitswerten. Somit ist es schwierig bis unmöglich, in einem Clustering-Verfahren einen sinnvollen Schnitt zwischen relevanten und irrelevanten Ereignissen zu unterscheiden. Die Gefahr ist groß, dass seltene aber effiziente Prozesspfade dabei gelöscht werden. Das Konzept der Aggregation hingegen ist zu den besonderen Eigenschaften von Case Management-Prozessen kompatibel. Durch Aggregation zusammengefasste Mengen von Ereignissen – beispielsweise, weil sie häufig gemeinsam auftreten – lassen sich komplizierte Zusammenhänge einfacher verstehen, ohne dass dabei zwangsläufig wichtige Informationen verloren gehen. Das adCM-Metamodell unterstützt diese Form von Komplexitätsreduktion durch die Möglichkeit zur Hierarchisierung der Agenda, jedoch ohne dass dabei Informationen über den Kontrollfluss der aggregierten Ereignisse benötigt werden.

In der Literatur lassen sich auch Metriken finden, welche die Vollständigkeit von Ereignisprotokollen errechnen beziehungsweise schätzen. Beispielsweise haben Yang et al. eine Metrik vorgestellt, die das Auftreten neuer Klassen von Ereignissen berücksichtigt (Yang et al. 2012). Hierfür werden zunächst Traces mit ähnlichem Verhalten zusammengefasst und dann untersucht, wie viele neue Klassen im Verlauf der Zeit hinzukommen. Dieser Zusammenhang lässt sich in einer Art Sättigungskurve darstellen, die eine Schätzung der Summe der Ereignisklassen erlaubt. Bezogen auf die Anwendung auf schwach strukturierte Prozesse kann eine der von den Forschern getroffenen Annahmen allerdings nicht gewährleistet werden: Die Anzahl ähnlicher Traces wird aufgrund des beinahe unerschöpflichen Variantenreichtums dieser Prozesse mit hoher Gewissheit nicht final bestimmbar sein. Der in dieser Dissertation vorgestellte

Lösungsansatz benötigt eine solche Einschränkung nicht, weil die Fälle nicht in ähnlichen Klassen zusammengefasst werden.

Unabhängig vom konkreten Anwendungsgegenstand des Ereignisprotokolls finden sich auch Metriken, die allgemein zur Errechnung von Komplexität im Sinne von Vielfalt herangezogen werden können. Hierzu zählen nach (van der Aalst 2011) der Gini-Index, erstmalig veröffentlicht in (Gini 1912) und die Entropie, einem Maß aus der Informationstheorie zur Berechnung des mittleren Informationsgehalts einer Nachricht (Shannon 2001). Entropie ist allerdings so definiert, dass sie eine *absolute* Aussage über die Vielfalt von Zeichen in einer Nachricht liefert. Im Gegensatz zu einem Maß relativ zur Länge der Nachricht wächst die Entropie also mit der Länge. Weil Traces in Ereignisprotokollen unterschiedlich lang sind, muss stattdessen die Entropierate herangezogen werden. Der Gini-Index hingegen skaliert zwischen 1 (maximale Vielfalt) und 0 (alle Elemente sind gleich). Beide Metriken sind aber so ähnlich, dass in der Data Mining-Literatur keine konkrete Empfehlung ausgesprochen werden konnte (Tan et al. 2005). Beispielsweise lassen sich beide auch einsetzen, um in den Ereignisprotokollen „Rauschen“ im Sinne von selten ausgeführten, gemäß traditionellem Process Mining irrelevanten, Aktivitäten zu identifizieren.

Bezugnehmend auf das im Process Mining-Manifest definierte Qualitätsmodell für Ereignisprotokolle (van der Aalst et al. 2012) lässt sich die Frage stellen, inwiefern sich die darin beschriebenen Anforderungen an die Ereignisprotokollqualität auf eine zu hohe Komplexität beziehen und daher ggfs. wiederverwendet oder weiterentwickelt werden können. In diesem Fall könnte das Modell durch bloße Anreicherung geeigneter Metriken als Rahmenwerk zur vergleichbaren Messung von Ereignisprotokollqualität eingesetzt werden, welches auch den Komplexitätsaspekt einschließt. Leider finden sich in den Qualitätsanforderungen nur Aspekte wie die Vollständigkeit, Vertrauenswürdigkeit oder Anforderungen an den Datenschutz und die Datensicherheit bei der Erzeugung, Speicherung und Verarbeitung der Protokolldateien.

Abschließend sei darauf hingewiesen, dass sich die in diesem Lösungsbaustein vorgestellten Metriken zur Messung von Ereignisprotokollkomplexität unmittelbar ergänzend einordnen lassen in das von Rozinat et al. entwickelte Process Mining Algorithm Evaluation Framework (Rozinat et al. 2008).

5.3 LÖSUNGSKONZEPT

Bei der Anwendung des Process Mining auf schwach strukturierte Prozesse besteht ein wesentlicher Teil der Methode darin, die erstellten Protokolle aus den vielen unterschiedlichen

Informationssystemen wieder in ein einheitlich strukturiertes Protokoll zusammenzuführen. Außerdem müssen die Ereignisdaten auf eine gemeinsame Abstraktionsebene gebracht (van der Aalst et al. 2012) und hierzu bei Bedarf zu höherwertigen Ereignissen zusammengefügt werden (Rath 2010). Letzteres ist insbesondere dann notwendig, wenn die Ereignisse nicht auf einer fachlichen Ebene („Gutachten lesen“), sondern auf einer technischen Ebene protokolliert werden („Dokument ‚Gutachten.docx‘ geöffnet“). Deshalb enthält das adCM-Konzept einen anderen Ansatz, um qualitativ hochwertige Ereignisse von Case Management-Prozessen zu protokollieren (Anforderungen 9 und 10).

Der einzige im Vorfeld bekannte Ort, an dem die Ereignisdaten in variantenreichen Prozessen gesammelt werden können, ist das Arbeitsumfeld des Case Managers selbst. Das adCM-Konzept sieht deshalb vor, genau an dieser Stelle ein **anwendungsübergreifendes Ereignisprotokoll** je Fall zu pflegen. Um dies zu erreichen, soll im Arbeitsumfeld des Case Managers ein zentrales Werkzeug mit einer entsprechenden **Monitoring-Komponente** eingesetzt werden. Das Werkzeug, dessen Architektur und Implementierung in Kapitel 7 beschrieben wird, ist außerdem dafür verantwortlich, die Ereignisse ohne zusätzliche Verfeinerungsschritte bereits in einer hochwertigen Qualität zu speichern. Um dies zu gewährleisten, soll das in Kapitel 4.3 vorgestellte Metamodell für die Typisierung der Ereignisse verwendet werden. Voraussetzung ist, dass dem Case Manager auf allen seinen Arbeitsgeräten dieses zentrale Arbeitsumfeld geboten wird. In diesem Kontext lassen sich dann die Aktionen in einem zentralen Ereignisprotokoll mitverfolgen und die Schritte zur Erzeugung, Extraktion, Filterung und Integration von unterschiedlichen Ereignisprotokollen fallen weg. Im weiteren Verlauf wird das Konzept zur zentralen Ereignisprotokollierung mithilfe des adCM-Metamodells beschrieben.

5.3.1 DATENMODELL FÜR DIE ANWENDUNGSÜBERGREIFENDE EREIGNISPROTOKOLLIERUNG

Analog zum Vorgehen beim Process Mining soll der Zustand domänenspezifischer Geschäftsobjekte protokolliert werden. In strukturierten Prozessen handelt es sich dabei zumeist um die Prüfung von Aufträgen oder das Ausstellen von Rechnungen. Entsprechend stehen in adCM-unterstützten Prozessen die Entitäten des Case Managements und deren Zustandsveränderungen im Mittelpunkt: Das Starten/Pausieren/Wiederaufnehmen und Beenden eines Falls, das Aufstellen und ggfs. Umsortieren einer Agenda und das Erledigen von Agendaeinträgen. Zusätzlich wird die Arbeit mit Artefakten protokolliert, sodass auch Aussagen über verwendete Datenquellen getroffen werden können.

Als Ergebnis dieser Überlegungen sieht das adCM-Lösungskonzept die folgenden basalen Entwurfsentscheidungen vor:

- anwendungsübergreifend einheitliches Ereignisprotokoll-Datenmodell
- die zentrale Protokollierung in einem wissensarbeitunterstützenden Werkzeug

Das Modell beantwortet die zu Anforderung **9** gehörenden Fragestellungen, welche Ereignisse protokolliert werden müssen. Hinsichtlich der Protokollqualität (Anforderung **10**) sorgt das Modell außerdem für eine **vollständige** Protokollierung, weil alle eingangs identifizierten Entitäten der Wissensarbeit sowie des adCM-Metamodells darin vertreten sind. Mithilfe der anwendungsübergreifenden Speicherung der Ereignisdaten in einem zentral dafür vorgesehenen Werkzeug wird zudem die **automatische** und **systematische** Protokollierung sichergestellt. Um auch der Forderung nach einer **wohldefinierten Struktur** nachzukommen, baut das Datenmodell auf der grundlegenden Struktur des XES-Standards (Extensible Event Stream) auf und ist dazu kompatibel (Verbeek et al. 2011). Der XES-Standard wurde 2010 von der IEEE Process Mining Task Force entwickelt und dient der interoperablen Speicherung von Ereignisdaten aus Anwendungen. Er berücksichtigt bereits die eingangs genannten Anforderungen an die semantische Anreicherung von Ereignisdaten, indem er die Möglichkeit bietet, die Standard-Nomenklatur für benutzerdefinierte Ereignisattribute zu erweitern und den Attributen damit eine klare Semantik zuzuweisen. Das Datenmodell ist in Abbildung 19 dargestellt. Seine Entitäten und Beziehungen werden im weiteren Verlauf erläutert.

Zu jedem Geschäftsprozess (zum Beispiel dem Reha-Management in der gesetzlichen Unfallversicherung) gibt es genau ein Protokoll. Dieses Protokoll enthält für jeden Fall (zum Beispiel der Rehabilitation von Herrn Schmidt) genau einen protokollierten Pfad durch den Prozess: dem Trace. Somit enthält ein Protokoll alle Pfade eines Prozesses. Die Pfade wiederum sind eine Kette von Ereignissen. Jedes Ereignis bezieht sich immer auf Entitäten des adCM-Metamodells. Sie bilden in diesem Zusammenhang auch die Menge an Ereignisquellen. Je nach Quelle können unterschiedliche vordefinierte Ereignisse protokolliert werden (zum Beispiel „gestartet“ oder „erledigt“). Die Farben der Ereignistypen kennzeichnen den Zusammenhang zwischen den Ereignissen und den Kernaktivitäten eines Wissensarbeiters. Im nächsten Abschnitt werden die Ereignisse im Detail vorgestellt und motiviert.

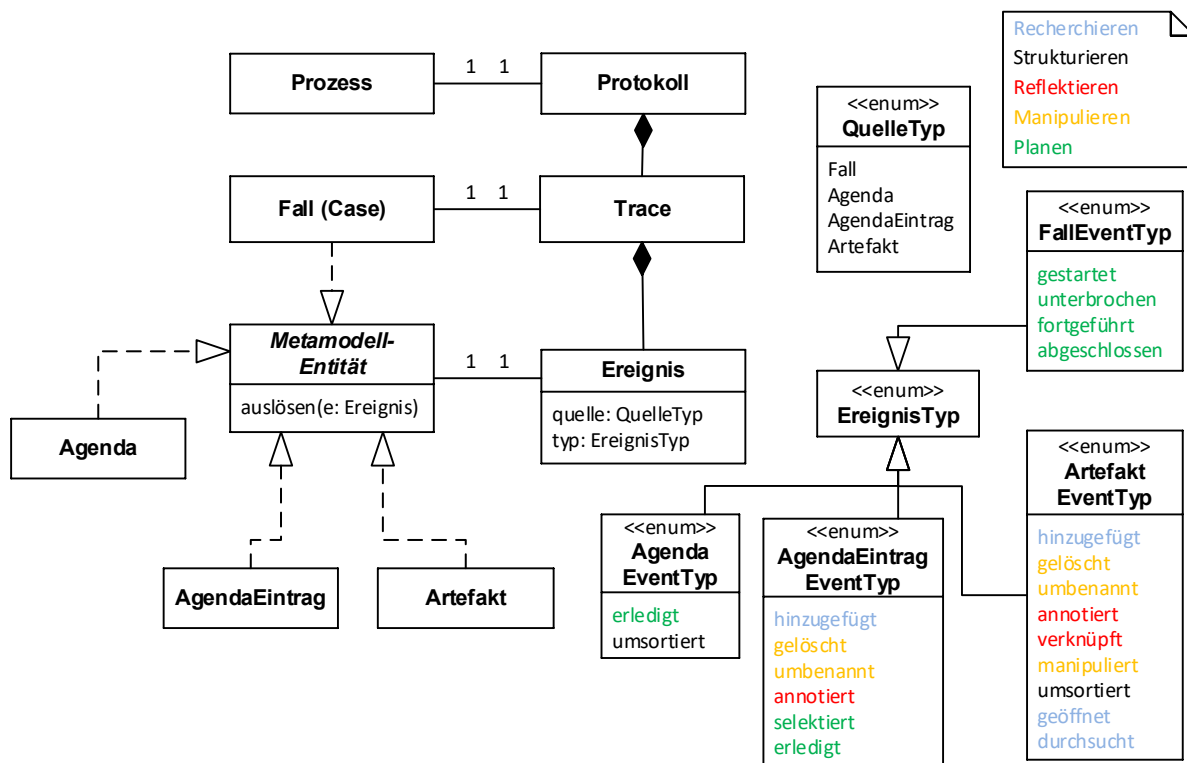


Abbildung 19: Datenmodell für die Ereignis-Protokollierung während der Arbeit mit Artefakten und Agenda

Die Agenda als Ganzes, das heißt der gesamte Baum aus Agendaeinträgen, kann als erledigt markiert werden. Zudem gibt es ein Ereignis speziell für die Agenda, welches ausgelöst wird, sobald die Elemente in der Agenda durch den Case Manager umstrukturiert werden (zum Beispiel, wenn zwei Agendaeinträge hinsichtlich ihrer Reihenfolge oder ihrer Ebene vertauscht werden).

Die Interaktionen des Case Managers mit Agendaeinträgen sind den Interaktionen mit Artefakten sehr ähnlich. Deshalb gibt es eine starke Überschneidung der Ereignistypen. So können Agendaeinträge und Artefakte gleichermaßen hinzugefügt, gelöscht, umbenannt oder mit einer Annotation versehen werden.

Im Process Mining spielt der Lebenszyklus einer Aktivität eine wichtige Rolle. Demnach wird beispielsweise zu jeder Aktivität festgehalten, ob es sich um den Start oder das Ende einer Aktivität handelt. Mithilfe dieser Zusatzinformationen kann unter anderem die Dauer einer Aktivität ermittelt werden. Weil Agendaeinträge gemäß ihrer Definition auch Aktivitäten repräsentieren können, wurde dieser Lebenszyklus mithilfe entsprechender Zustände („selektiert“ und „erledigt“) ebenfalls implementiert. Das Selektieren eines Agendaeintrags entspricht dem Fortführen der damit verbundenen Aktivität. Ist diese Aktivität hingegen abgeschlossen, wird der Agendaeintrag als erledigt markiert. Ein ähnliches Lebenszyklusmodell existiert auch für den übergeordneten Fall und die Agenda als Ganzes. Ein Fall kann „gestartet“,

„unterbrochen“, „fortgeführt“ und „abgeschlossen“ werden. Ein Fall kann beliebig oft unterbrochen und fortgeführt, aber nur einmal gestartet und abgeschlossen werden.

Neben den bereits oben beschriebenen Interaktionen des Case Managers mit Artefakten, wird auch das Verknüpfen von Artefakten untereinander protokolliert. Sofern der Case Manager diese Funktion des Workspaces entsprechend nutzt, kann auf diese Weise mitverfolgt werden, wie sich semantische Netze im Verlauf eines Falls auf den Workspaces bilden. Zusätzlich dazu wird auch das Öffnen eines Artefakts protokolliert. Werden bestimmte Artefakte besonders häufig geöffnet, kann mithilfe des Protokolls auf die Bedeutung dieser Artefakte geschlossen werden. Um zwischen Eingangs- und Ausgangsartefakten differenzieren zu können, werden darüber hinaus auch alle Veränderungen an einem Artefakt mithilfe des Ereignisses „manipuliert“ festgehalten. Werden beispielsweise bestimmte Artefakte häufig geöffnet, aber nicht oder nur selten manipuliert, so handelt es sich sehr wahrscheinlich um Artefakte, die der Case Manager nicht selbst erstellt beziehungsweise pflegt, sondern vielmehr als Informationsquelle für seine Aufgaben einsetzt. Ein ähnlicher Zusammenhang ist zu erwarten, wenn der Case Manager ein Artefakt häufig nach Stichworten durchsucht. Deshalb wird auch das Durchsuchen eines Artefakts protokolliert. Das Umsortieren von Artefakten sowohl innerhalb als auch zwischen verschiedenen Workspaces wird jeweils als Ereignis festgehalten, um die Kernaktivität „strukturieren“ zu beobachten.

Um das Verständnis dieser Zusammenhänge zu erleichtern, wird die Verwendung der im Modell definierten Ereignistypen an einem Beispiel-Protokoll gezeigt (siehe Tabelle 1).

Tabelle 2: Beispiel einer Protokoll-Datei (Prozess "Lehrveranstaltung planen und durchführen")

#	Zeit-stempel	Quelle	Ereignistyp	Ereignisspezifische Attribute
1	578921	Fall	gestartet	„id:4711“
2	578921	Agenda	ausgewählt	„id:0“, „name:Trainer aussuchen“
3	578942	Artefakt	geöffnet	„ort:wiki/userprofiles“
4	579230	Artefakt	durchsucht	„ort:wiki/userprofiles“, „begriffe:..."
5	579454	Artefakt	hinzugefügt	„src:wiki/userprofiles/EHE“, „dst:<current_workspace>“
6	579456	Agenda	erledigt	„id:0“, „descr: Trainer aussuchen“
7	579456	Agenda	ausgewählt	„id:1“, „descr: Trainer informieren“
8	579469	Agenda	ausgewählt	„id:2“, „Kurstermine und -Orte abstimmen“
9	580510	Artefakt	verknüpft	„src:{Outlook-Terminobjekt}“, „dst:{Agenda-Eintrag, id:2}“
10	580521	Agenda	erledigt	„Kurstermine und -Orte abstimmen“
...

75	580521	Agenda	erledigt	„id:22“, „descr: Zertifikate drucken“
76	580521	Fall	abgeschlossen	„id:4711“

Das Protokoll gehört zu einem Beispiel-Prozess „Lehrveranstaltung planen und durchführen“. Es beginnt mit einem Trace, der alle Benutzerinteraktionen zum Fall 4711 enthält. Der erste Agenda-Eintrag „Trainer aussuchen“ wird automatisch vom System ausgewählt. Eine kurze Zeit später öffnet der Case Manager die Intranet-Seite mit den Benutzerprofilen der verfügbaren Dozenten. Er sucht darin nach Spezialisten zum Thema „Spezifikation“ und zieht dann das Profil eines Dozenten mit dem Kürzel „EHE“ auf den Workspace. Er markiert den Agendaeintrag als „erledigt“ und bevorzugt entgegen des Vorschlags vom System den Eintrag „Kurstermine und -Orte abstimmen“. Eine ganze Weile später – vermutlich mussten komplizierte Absprachen getroffen werden – wird ein Outlook-Terminobjekt auf den Agendaeintrag gezogen. Unmittelbar danach markiert der Case Manager den Eintrag als „erledigt“. Mit dem Druck der Zertifikate wurde der letzte offene Agendaeintrag als „erledigt“ markiert und das System schlägt den Abschluss des Falls vor.

5.3.2 MONITORING MIT SEMANTISCH VERKNÜPFTEN EREIGNISSEN

Nach Aalst et al. sollte ein qualitativ hochwertiges Protokoll Ereignisse enthalten, deren Eigenschaften mit einer Ontologie verknüpft werden. Auf diese Weise kann die Semantik der Eigenschaften nicht nur von Menschen, sondern auch von Maschinen verstanden werden, was wiederum neue Möglichkeiten für die Auswertung der Ereignisdaten liefert. In dem folgenden Abschnitt wird beschrieben, mit welchem Lösungskonzept diese Verknüpfung im Agenda-driven Case Management hergestellt wird.

Sobald ein Ereignis eingetreten ist, wird es nicht nur in einer XES-kompatiblen Protokolldatei festgehalten. Das Ereignis wird zusätzlich an eine Serverkomponente (*Monitoring-Dienste*) geschickt, die für die semantische Verknüpfung der Ereignisinformationen mit den adCM-Konzepten aus dem Metamodell verantwortlich ist (siehe Abbildung 20). Mit einer solchen Verknüpfung ist zunächst nichts Anderes gemeint als die Typisierung von Informationen in den Ereignissen auf der Grundlage von in der Ontologie definierten Typen. Am Beispiel von adCM müsste also ein Ereignis „Agenda hinzugefügt“ mit dem Typ „Agenda“ aus dem adCM-Metamodell verknüpft werden. Damit realisiert die Komponente das noch fehlende Qualitätsmerkmal aus der Anforderung 10, dass die Attribute eine **eindeutige und bekannte Semantik** haben sollen. Die Komponente transformiert die sequentiell eintreffenden Ereignisse in ein Objektmodell, das den aktuellen Zustand des betroffenen Falls repräsentiert. Das Objektmodell wird in einer

Wissensdatenbank vorgehalten und dort für weitere Anwendungsfälle zu nachfolgenden Lösungsbausteinen bereitgehalten.

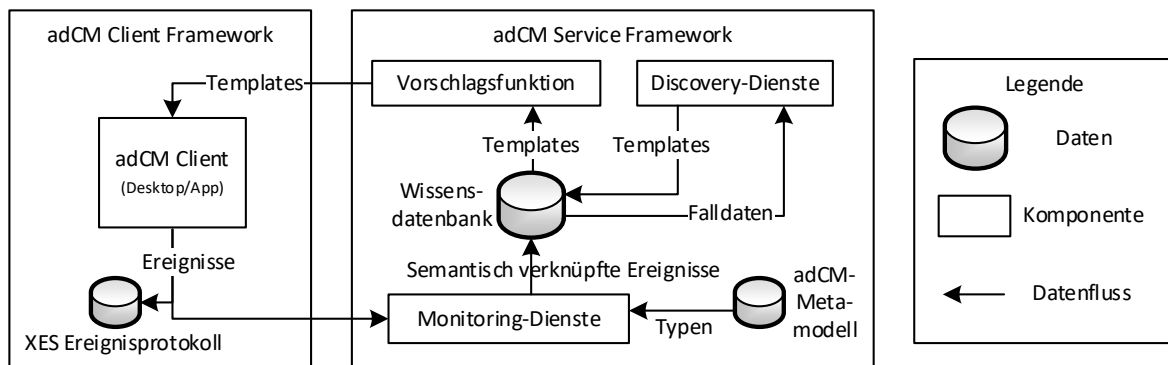


Abbildung 20: Facharchitektur für die semantische Verknüpfung

Für das Objektmodell in der Wissensdatenbank wird eine Datenstruktur benötigt, die für Maschinen gut lesbar ist und möglichst flexibel neue, möglicherweise aus bestehenden Informationen geschlussfolgerte Verknüpfungen aufnehmen kann. In den nachfolgenden Absätzen wird im Detail erläutert, wie diese Datenstruktur Stück für Stück aufgebaut wird.

Ereignisprotokolle werden im Process Mining dafür eingesetzt, sequenzielle Abhängigkeiten zu identifizieren und diese Abhängigkeiten in Prozessmodelle zu übertragen. Allerdings haben Ereignisprotokolle auch einen klaren Nachteil: Mit ihnen lassen sich nicht besonders gut die aktuellen Zustände der bearbeiteten Fälle ablesen. Das ist aber notwendig zur Identifizierung des aktuellen Kontexts eines Falls (siehe auch Kapitel 5.3.3), zum Beispiel, um passende Artefakte oder Agendaeinträge vorschlagen zu können. Dass man diese Informationen nicht unmittelbar aus einem Protokoll ablesen kann, liegt nicht nur daran, dass man diese Zustände aus den einzelnen Ereignissen rekonstruieren müsste. Vielmehr liegt das Problem darin, dass die Ereignisse kaum Informationen über die Zustände der verarbeiteten Objekte enthalten, sondern lediglich über die jeweilige Aktivität, mit der dieser Zustandswechsel hervorgerufen wurde.

Um die fehlenden Informationen in der Wissensdatenbank zusammenzutragen, werden die Ereignisse unmittelbar nach der semantischen Typisierung für die Erzeugung und Fortschreibung eines Modells aus adCM-Objekten eingesetzt. Der zugrundeliegende Algorithmus wird anhand eines Beispiels erläutert: Wenn der Case Manager Herr Miller einen neuen Fall zur Rehabilitierung des Patienten Rob Smith startet, wird zunächst durch das dafür benutzte zentrale adCM-Werkzeug ein Ereignis „Fall gestartet“ erzeugt und die (noch leere) Agenda selektiert. Beide Ereignisse werden dann in der XES-Datei protokolliert und an eine serverseitige Komponente geschickt. Diese leitet das Ereignis zunächst ungefiltert weiter zu Speicherung in eine

graphbasierte Wissensdatenbank (die Auswahl dieser Technologie wird in Kapitel 6.3 motiviert). Dort liegen beide Ereignisse als Elemente einer verketteten Liste von Ereignissen, die im weiteren Verlauf des Falls ergänzt wird, zum Beispiel durch das Hinzufügen eines neuen Agendaeintrags. Vorgehend auf die in Kapitel 7 enthaltenen Implementierungsdetails zeigt Listing 1 diese verkettete Liste in N3 Turtle-Notation. Die ersten drei Zeilen sind Präfix-Definitionen. Sie ermöglichen eine verkürzte Schreibweise der URLs, über die die einzelnen Elemente eindeutig identifiziert werden. Mit dem ersten Präfix beginnen alle Elemente des adCM-Metamodells sowie des adCM-Ereignisdatenmodells. Mit dem zweiten Präfix beginnen alle Elemente, die zu den Beispiel-Falldaten gehören und das letzte Präfix verweist auf alle Elemente der graphbasierten Sprache RDF. Das erste Ereignis der Liste wurde erzeugt durch das Starten des Falls (Ereignis „CaseStarted“) mit dem Namen „Rehab Rob Smith“. Das zweite Ereignis wurde erzeugt durch die Auswahl der zu diesem Zeitpunkt noch leeren Agenda. Weil es gemäß des adCM-Metamodells pro Fall nur genau eine Agenda gibt, trägt sie den Namen des Falls. Das letzte Ereignis ist durch das Hinzufügen eines neuen Agendaeintrags (Ereignis „AgendaItemAdded“) mit dem Namen „create healing plan“ ausgelöst worden. Die drei Elemente bilden eine verkettete Liste über die Eigenschaft „adcm:nextEvent“, die jeweils auf die URL des nachfolgenden Ereignisses zeigt.

Präfixe
(zur besseren Lesbarkeit)

```

@prefix adcm: <http://paluno.uni-due.de/adcm#> .
@prefix ex: <http://paluno.uni-due.de/example/> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .

```

Ereignisse

```

ex:Event/7a549db6-ed59-45dd-83dd-9f0015cc84e3
  rdf:type          adcm:CaseStarted ;
  adcm:timestamp    2014-06-02T13:15:37.650Z ;
  adcm:nextEvent    ex:Event/30d46ffd-f874-4487-a426-3dab6de0e612 ;
  adcm:sourceId     "Rehab Rob Smith" .

ex:Event/30d46ffd-f874-4487-a426-3dab6de0e612
  rdf:type          adcm:AgendaSelected ;
  adcm:timestamp    2014-06-02T13:15:37.870Z ;
  adcm:nextEvent    ex:Event/875a91af-1360-4e3c-8bae-a5f645388b52 ;
  adcm:sourceId     "Rehab Rob Smith" .

ex:Event/875a91af-1360-4e3c-8bae-a5f645388b52
  rdf:type          adcm:AgendaItemAdded ;
  adcm:timestamp    2014-06-02T13:15:39.540Z ;
  adcm:sourceId     "create healing plan" .

```

Ereignis-ID

Ereignistyp

Datum/Uhrzeit

Verweis auf das nächste Ereignis

ID der konkreten Ereignisquelle

Listing 1: Rohe Eventdaten in der Graphdatenbank (in N3 Turtle-Notation)

Sobald ein neues Ereignis in dieser Liste gespeichert wird, muss zuerst der als Zeichenkette angegebene Typ ausgelesen und dem passenden adCM-Konzept aus dem adCM-Metamodell zugeordnet werden. Anhand dieser Typinformation wird anschließend jeweils das entsprechende adCM-Objekt instanziiert (siehe Abbildung 21). Nachdem der Case Manager einen neuen Agendaeintrag namens „create healing plan“ hinzugefügt hatte, wurde ein neues Ereignis der Liste hinzugefügt und anschließend ein Agendaeintrags-Objekt erzeugt und der noch leeren Agenda hinzugefügt. Ein paar Tage später ist der Heilverlaufsplan tatsächlich erstellt worden und der Case

Manager markiert den Agendaeintrag als „erledigt“. Auch diese Aktion wird durch das in Kapitel 5.3.1 vorgestellte Ereignisdatenmodell abgebildet. Folglich wird das entsprechende Ereignis erzeugt und an die Serverkomponente geschickt, die es wiederum in der Liste ergänzt und die Zustandsänderung im Objektmodell vornimmt (was in der Abbildung durch ein Durchstreichen des Agendaeintrags symbolisiert wird). Während des Rehabilitationsprozesses von Herrn Smith folgen viele weitere Zustandsveränderungen am Objektmodell des Falls. Nach ein paar Wochen ist der Fall abgeschlossen und alle Agendaeinträge, einschließlich des Wurzelements, sind als erledigt markiert worden. Die Wissensdatenbank enthält nun einen konsistenten Bestand an detaillierten Informationen über diesen Fall. Mithilfe des Objektmodells ist ein schneller Zugriff auf den finalen Zustand des Falls möglich. Will man Informationen darüber erhalten, wie sich dieser Fall über die Zeitachse entwickelt hat und in welcher Reihenfolge die Aktivitäten aus der Agenda abgeschlossen wurden, kann auf die Ereignisliste zugegriffen werden. Prinzipiell ist es dadurch sogar möglich, den Fall im Rahmen von Analyseverfahren vor- und zurückzuspulen.

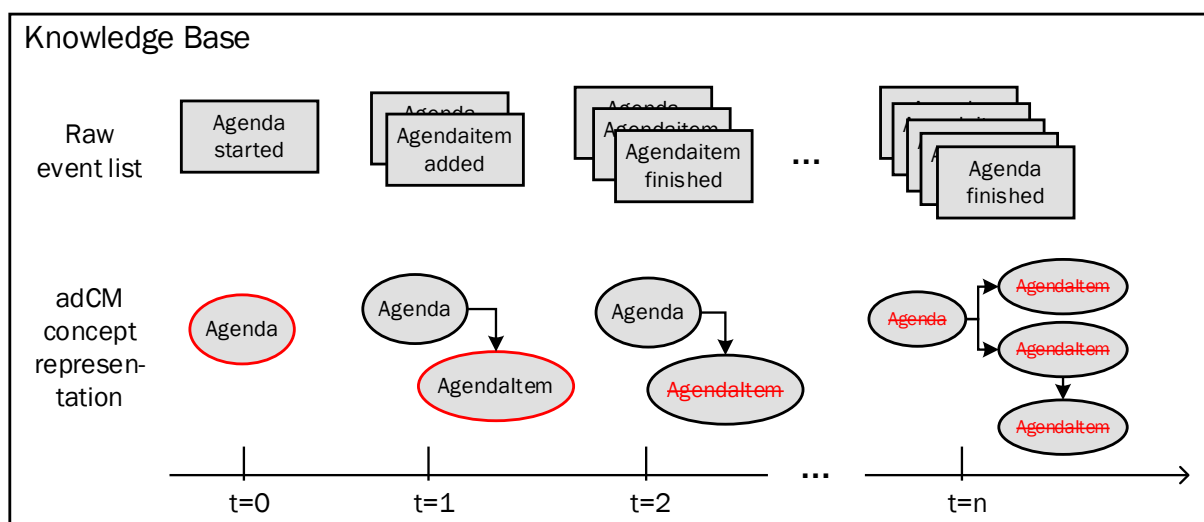


Abbildung 21: Aufbau der Objekt-Datenstruktur zu einem Fall mithilfe der Ereignisse

Damit die Algorithmen der nachfolgenden Discovery-Schritte stets den Zusammenhang zwischen den Ereignissen und den adCM-Objekten herstellen können, sind beide Datenstrukturen miteinander verknüpft. Das heißt, jedes Element aus der Ereignisliste verfügt über eine Objektreferenz zu dem betreffenden Element aus dem adCM-Objektmodell. Filtert man beispielsweise alle Ereignisse zu einem Objekt heraus, kann man sehr genau die Historie dieses einen Objekts rekonstruieren.

An dieser Stelle sei allerdings angemerkt, dass es sich bei den oben genannten Schritten nur um unterschiedliche Repräsentationen des im Rahmen des Monitorings beobachteten Prozessverlaufs handelt. Es sind noch keine zusätzlichen Informationen in die Wissensdatenbank

hineingeflossen, zum Beispiel durch logische Schlussfolgerungen oder fallübergreifende Aggregationen von Prozesswissen. Solche weiterführenden Schritte sind dem Discovery zuzuordnen und werden in Kapitel 6 vorgestellt.

5.3.3 *MONITORING DES FALL-KONTEXTS*

Die in dieser Dissertation aufgeführten Lösungsbestandteile dienen dem Zweck, Wissensarbeit durch gesammeltes Prozesswissen in Gestalt von Templates zu unterstützen. Templates sind aber so definiert, dass sie nur in bestimmten Kontexten sinnvoll sind. Gemäß der Anforderung **12** soll deshalb im Rahmen des Monitoring-Lösungsbausteins eine Lösung dafür gefunden werden, was als Kontextinformation eines Falls infrage kommen könnte und wie diese Kontextinformationen zur Laufzeit an die nachgelagerten Lösungsbausteine weitergereicht werden können. Demzufolge beginnt dieser Abschnitt mit einer kurzen Abhandlung über den Kontext, inklusive einer Definition des Kontextbegriffs.

Zweck des Kontexts

Um einen geeigneten Kontextbegriff zu formulieren, ist es hilfreich, sich vorab genauer mit dem Zweck dieses Kontextes zu befassen. Im Agenda-driven Case Management sind in den nachfolgenden Lösungsbausteinen neben der Template-Vorschlagsfunktion auch Unterstützungsfunktionen für das Recherchieren in heterogenen Datenquellen wie unternehmensinterne Datenbanken oder Dokumentablagen vorgesehen. Sie sollen zum aktuellen Kontext passende Artefakte vorschlagen beziehungsweise die Suchergebnisse derart umsortieren, dass solche Artefakte vorrangig angezeigt werden. Es dreht sich also unter anderem um das aktuelle Informationsbedürfnis des Case Managers. Wenn er zum Beispiel gerade die Diagnose eines Reha-Patienten mit einer Auflistung von Symptomen liest und anschließend nach einem Facharzt sucht, sollten die Informationen in der Diagnose bei der Auswahl des Suchergebnisses berücksichtigt werden.

In der Information Retrieval-Community (Baeza-Yates 2015), bei der es ebenfalls um wissenschaftliche Beiträge zum Auffinden kontextrelevanter Informationen aus ansonsten für Benutzer verborgenen Datenquellen geht, kann nur auf sehr wenig Informationen über den Kontext zurückgegriffen werden (Köhncke et al. 2013). Denn im einfachsten Fall handelt es sich schlicht um die Begriffe in der Suchabfrage. Etwas kompliziertere Ansätze verwenden kürzlich vorangegangene Suchbegriffe oder den Titel und Inhalt offener Browsertabs, um die Suchergebnisse zu verbessern (Kraft et al. 2006). Im adCM ist diese Herangehensweise nicht vielversprechend. Das liegt an einer Kombination zweier Sachverhalte: Erstens arbeiten Case

Manager nicht selten parallel an verschiedenen Fällen oder wechseln zumindest öfters zwischen einzelnen Fällen hin und her. Zweitens dient der Browser als Portal für alle möglichen Informationsbedürfnisse, sodass insbesondere in der Wissensarbeit nicht auszuschließen ist, dass geöffnete Tabs und alte Suchabfragen zu einem völlig anderen Kontext gehören, als der aktuell geöffnete Tab. Der Kontextbegriff muss also so definiert sein, dass er möglichst frei von solchen Störungen ist.

Nicht nur der Blick auf die Artefakte ist hilfreich, um sich dem Kontextbegriff zu nähern. Denn darüber hinaus ist es wahrscheinlich, dass der Case Manager wichtige Informationen über den Patienten auch in den Agendaeinträgen festhält. Wenn er beispielsweise gerade den Agendaeintrag „Facharzt für Oberschenkelhalsfrakturen konsultieren“ selektiert hat, könnten ebenfalls dazu passende Suchergebnisse vorgezogen werden. Neben dem Vorschlagen von Templates und Artefakten gibt es einen weiteren wichtigen Anwendungsfall, bei dem der Kontext eine wichtige Rolle spielt: der Einsatz von Geschäftsregeln. Bei Hospitationen im Reha-Management wurde von einer Klassifikation von Fällen berichtet, die für das Controlling wichtig ist. Eine solche Klassifikation automatisch anhand des Kontextes und eines Regelwerks identifizieren zu können, wäre demnach vorteilhaft.

Kontext-Definition

Ausgehend von den verschiedenen Zwecken, denen die Identifikation des Kontexts dienen soll, wurde das adCM-Metamodell mit dem Ziel untersucht, potentielle Informationsquellen für einen solchen Kontext zu finden. Im Ergebnis wurden drei Möglichkeiten abgeleitet. Je nachdem, welcher konkrete Zweck mit dem identifizierten Kontext verfolgt werden soll, kann der passende Kontextbegriff gewählt werden:

- **Kontext des Falls:** Der Kontext eines Falls wird durch das Objektmodell beschrieben. Somit sind alle Agendaeinträge und Artefakte inklusive ihrer Beziehungen zueinander Teil des Kontexts.
- **Kontext der Aktivität:** Der Kontext besteht aus der aktuell selektierten Aktivität, zusammen mit allen unmittelbar untergeordneten Aktivitäten und Artefakten.
- **Einbettung der Aktivität:** Der übergeordnete Kontext einer Aktivität besteht aus dem Pfad übergeordneter Agendaeinträge, bis zum Wurzelement der Agenda.

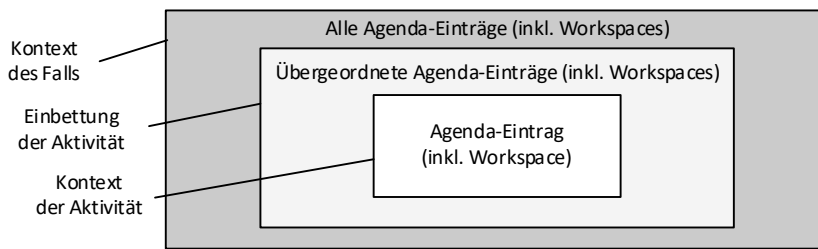


Abbildung 22: adCM Kontext-Hierarchie

Im weiteren Verlauf soll die Auswahl des passenden Kontextbegriffs am Beispiel der Artefakt-Vorschlagsfunktion gezeigt werden.

Der Workspace und alle darin enthaltenen Artefakte sind bereits vom Case Manager ausgewählt und dem Fall hinzugefügt worden, weil sie in unmittelbarem Bezug zu der Aktivität stehen. Andernfalls hätte er sie an anderer Stelle oder gar nicht dem Fall hinzugefügt. Somit repräsentieren diese Artefakte zu einem hohen Grad den aktuellen Kontext, aus dem heraus sich das neue Informationsbedürfnis bildet. Zugegebenermaßen würde man alle diese Artefakte auch in den Browserhistorien wiederfinden, jedoch wären dort auch viele irrelevante Dokumente enthalten. Die Filterung des Kontexts nimmt insofern der Case Manager während seiner Arbeit mit den adCM-Konzepten selbst vor. Weil aber die Namen der Artefakte selbst nicht aussagekräftig genug sind, sieht das Lösungskonzept vor, die in den Artefakten enthaltenen Informationen als Verfeinerung des Kontexts zu verwenden. Für diesen Anwendungsfall gibt es bereits gereifte Werkzeuge mit guten Ergebnissen aus der Computerlinguistik, insbesondere aus dem Forschungsfeld der Informationsextraktion. Sie dienen der Gewinnung einer Menge von Schlüsselwörtern, die eine möglichst hohe Informationsdichte haben und deshalb von sprachlich unwichtigen Begriffen (sogenannte Stopwords) bereinigt wurden. Diese Gründe führen zu der Entscheidung, für die Artefakt-Vorschlagsfunktion den Kontext der Aktivität als Kontextbegriff zu wählen.

Bisherige Ansätze, wie beispielsweise die von Kraft et al. (2005), sind darauf angewiesen, Kontextinformationen manuell von den Case Managern zu erfragen. Diese Vorgehensweise wird beim adCM nicht benötigt, weil man die Strukturen aus Agendaeinträgen und Artefakten verwendet, die bereits vom Case Manager selbst im Rahmen seiner Arbeit hergestellt werden. Dadurch können ablenkende Zwischenfragen vom System vermieden werden.

Identifizierung des Kontexts

Um die benötigten Kontextinformationen aus dem Workspace zu erhalten, werden die erfolgreichen Ansätze aus (Kraft et al. 2005) adaptiert. Hierzu werden die Artefakte aus dem

Workspace durch sogenannte Context Term-Vektoren repräsentiert, die die wesentlichen Schlüsselwörter zu den einzelnen Dokumenten enthalten (siehe Abbildung 22).

Definition: Context Term-Vektor

Ein Context Term-Vektor ist gemäß dem Vector Space Model (Salton et al. 1975) ein n -Dimensionaler Vektor, der eine Suchabfrage oder ein Dokument repräsentiert. Der Vektorraum wird aufgespannt durch Elemente (Schlagwörter, Wörter oder Phrasen) eines analysierten Textes. Enthält eine Suchabfrage oder ein Dokument ein Element aus dem analysierten Text, so ist der entsprechende Wert des Vektors ungleich null. Mithilfe der linearen Algebra kann somit die Ähnlichkeit zwischen zwei Vektoren gemessen werden.

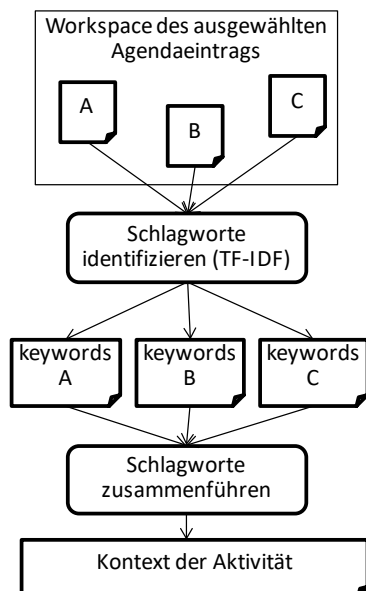


Abbildung 23: Erzeugung des Kontexts

Um diese Schlüsselwörter zu erhalten, muss der Text im Dokument zunächst durch einen sogenannten Tokenizer in einzelne Token zerlegt werden. Weil in den meisten Sprachen rein aus syntaktischen Gründen die Regeln für Groß- und Kleinschreibweise unterschiedlich sind (zum Beispiel beim Satzanfang), müssen alle Wörter anschließend in Kleinschrift umformatiert werden. Als nächstes werden alle Wörter entfernt, die in der gegebenen Sprache nicht bedeutungstragend sind. In der deutschen Sprache handelt es sich dabei zum Beispiel um bestimmte und unbestimmte Artikel sowie Konjunktionen. Die übrig gebliebenen Wörter werden im letzten linguistischen Schritt auf ihre Stammform reduziert, die durch verschiedene Deklinationen und Konjugationen hervorgerufen werden.

Nach diesen rein linguistischen Vorverarbeitungsschritten wird mithilfe statistischer Verfahren die Relevanz der Schlüsselwörter gemessen. Das passiert mit TF-IDF (term frequency-inverse document frequency), einer Standard-Metrik aus dem Information Retrieval. Nach Paik (2013) handelt es sich dabei um einen zuverlässigen Indikator für die Relevanz eines Begriffs in einem Dokument, das sich wiederum in einer Kollektion aus Dokumenten befindet. Diese Situation findet sich im adCM wieder, denn die ausgelesenen Artefakte befinden sich in einem gemeinsamen Workspace des aktuellen Agendaeintrags.

TF-IDF

Die TF-IDF-Metrik eines Wortes besteht aus zwei Metriken. TF (term frequency) gibt die Häufigkeit des Vorkommens eines Wortes innerhalb eines gegebenen Dokuments an (siehe Abbildung 24). Je nach Implementierung wird dieser absolute Wert auch normiert, beispielsweise gegen die maximalen Vorkommen über alle Wörter. IDF (inverse document frequency) beschreibt hingegen die Bedeutung eines Wortes bezogen auf eine Menge von Dokumenten. Im Gegensatz zu TF wird hier aber nicht jedes Vorkommen gezählt, sondern nur die Anzahl der Dokumente, in denen das Wort auftritt (df, document frequency). Dieser Wert wird ins Verhältnis zur Gesamtanzahl der Dokumente N gesetzt, um eine relative Kennzahl auf einem Intervall zwischen 0 und 1 zu erhalten. Die Bedeutung des Logarithmus in der Formel wird in (Robertson 2004) diskutiert.

Term frequency		Document frequency	
n (natural)	$tf_{t,d}$	n (no)	1
l (logarithm)	$1 + \log(tf_{t,d})$	t (idf)	$\log \frac{N}{df_t}$
a (augmented)	$0.5 + \frac{0.5 \times tf_{t,d}}{\max_t(tf_{t,d})}$	p (prob idf)	$\max\{0, \log \frac{N-df_t}{df_t}\}$
b (boolean)	$\begin{cases} 1 & \text{if } tf_{t,d} > 0 \\ 0 & \text{otherwise} \end{cases}$		
L (log ave)	$\frac{1 + \log(tf_{t,d})}{1 + \log(\text{ave}_{t \in d}(tf_{t,d}))}$		

Abbildung 24: Definitionen von TF und IDF (Manning et al. 2008)

Um TF-IDF zu erhalten, werden beide Metriken miteinander multipliziert (siehe Formel 1). Anhand des TF-IDF-Werts kann der Vektor nach Relevanz absteigend sortiert werden. Damit ist die Identifizierung des Kontexts abgeschlossen und der Kontext-Vektor kann von anderen adCM-Methoden zur Unterstützung der Wissensarbeit verwendet werden.

Formel 1: Definition von TF-IDF (Manning et al. 2008)

$$TF-IDF(t, d, D) = TF_{t,d} \times IDF(t, D)$$

5.3.4 MONITORING-ARCHITEKTURENTSCHEIDUNGEN

Ausgehend von dem oben beschriebenen Ansatz stellt sich die Frage, auf welche Weise diese Ereignisdaten erhoben werden können. Die resultierenden Architekturentscheidungen basieren auf den grundlegenden Ansätzen aus (Benner et al. 2012) und werden im weiteren Verlauf

verfeinert. Um einfach nachvollziehbare Szenarien zu erhalten, wird auch wieder auf die Kernaktivitäten Bezug genommen.

Um alle Benutzerinteraktionen im Rahmen der **Recherche**-Aktivitäten protokollieren zu können, muss zunächst das Problem gelöst werden, dass die recherchierten Artefakte an verschiedenen Orten gespeichert und mit verschiedenen Anwendungen assoziiert sind. Unternehmensinterne Daten, also Daten, die in Datenbanken, Dokumenten, E-Mails und Wikis/Intranets gespeichert sind, können mithilfe von Unified Information Access (UIA)-Technologien (Peinl 2011) erschlossen werden. Daher soll das Konzept auch eine entsprechende Schnittstelle vorsehen. Gestartete Suchanfragen werden dann an die Unternehmens-Suchmaschine sowie an Internet-Suchmaschinen weitergeleitet und die Ergebnisse werden gemeinsam visualisiert (siehe Abbildung 25).

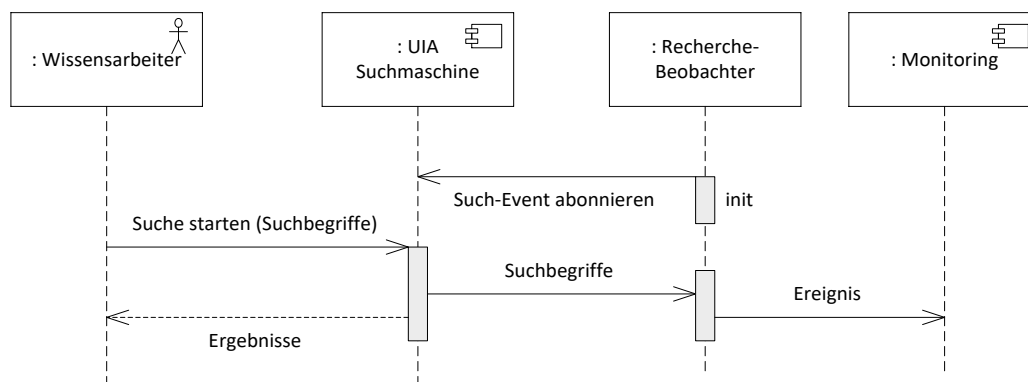


Abbildung 25: Protokollierung von Recherche-Aktivitäten an der zentralen UIA-Suchmaschine

Auch der Workspace bietet eine gute Grundlage für das Festhalten von **Strukturierungs**-Aktivitäten. Die Möglichkeiten, Artefakte auf demselben Workspace miteinander in Beziehung zu setzen, sind aber bislang nur rudimentär in Kapitel 4.3 beschrieben worden. Es fehlen Konzepte, wie den Beziehungen eine Semantik verliehen werden kann. Diese Informationen sind wichtig, um in späteren Schritten aus dem Protokoll heraus Muster und Zusammenhänge zwischen Artefakten erkennen zu können. Die Beziehungen sollten also bei Bedarf – ähnlich einer Ontologie – unterschiedlich getypet werden können, um dem Case Manager mehr Möglichkeiten für eine feinere Strukturierung von Artefakten zu bieten. Die Abbildung 26 zeigt ein mögliches Datenmodell zur Beschreibung solcher Beziehungen. Beziehungen zwischen Artefakten können eine Richtung haben und mit einer Semantik versehen werden (zum Beispiel A verfeinert B, A ist ein Teil von B, A verweist auf B, A ersetzt B, A definiert B, A bewertet B usw.). Das Konzept sieht zudem einen Standard-Pool an vordefinierten Beziehungen vor, der durch den Case Manager bei Bedarf erweitert werden kann.

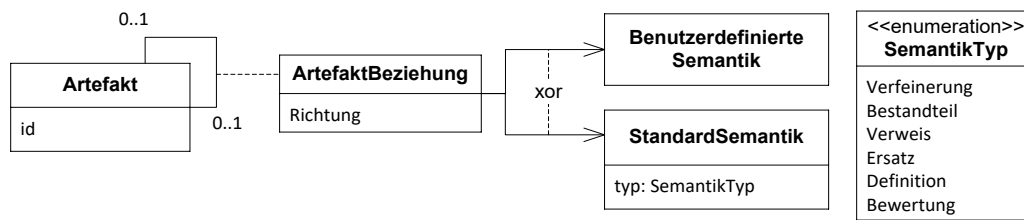


Abbildung 26: Datenmodell für die Beziehungen zwischen Artefakten

Um das **Manipulieren** von Artefakten zu protokollieren, kann – wie schon beim Recherchieren – auch der Browser hinzugezogen werden. Denn das manipulierte Artefakt, dessen Einbettung im Workspace sowie die Zuordnung zu einem Agenda-Eintrag sind bekannt und bilden einen Kontext, dem die entsprechenden Benutzerinteraktionen zugeordnet werden können. Andreas S. Rath hat im Rahmen seiner Dissertation (Rath 2010) bereits eine Methode entwickelt, wie aus dem Betriebssystem und aus Programmen heraus Manipulations-Ereignisse protokolliert werden können. Der dort formulierte Ansatz soll adCM in dieser Hinsicht ergänzen.

Interaktionen, die der Kernaktivität **Interpretieren** zugeordnet werden können, finden sowohl im Zusammenhang mit der Agenda als auch mit dem Workspace und den geöffneten Artefakten statt. Damit die dazugehörigen Benutzerinteraktionen zentral protokolliert werden können, muss zunächst überhaupt erst die vollständige Funktionalität sichergestellt werden. In diesem Zusammenhang entsteht eine Reihe von Herausforderungen: (1) der Umgang mit schreibgeschützten Artefakten, (2) fehlende technische Unterstützung für die Einbindung von Annotationen in der Dateistruktur, (3) unterschiedliche Mechanismen zur Einbettung von Annotationen innerhalb des Textes (Inline- oder per Verweis) sowie (4) die je nach Anwendung unterschiedliche User Experience, d. h. variierende Konzepte zur Darstellung und Pflege von Annotationen. Der hier verwendete Lösungsansatz sieht die Entwicklung eines anwendungsübergreifenden Datenmodells sowie einer Infrastruktur für Annotationen vor, mit dedizierten Komponenten zur Erstellung und Verwaltung von Annotationen (siehe Abbildung 27).

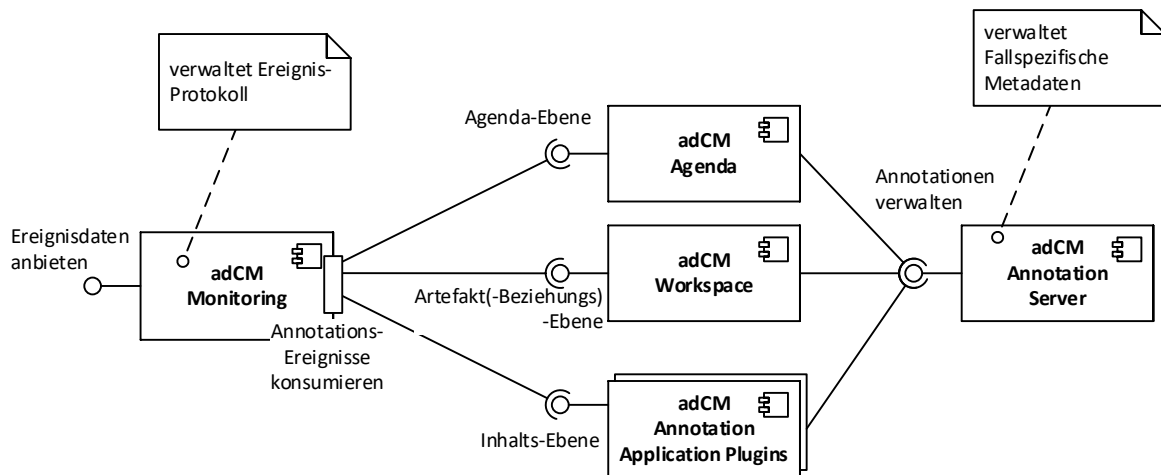


Abbildung 27: Infrastruktur zur Erstellung und Pflege von Annotationen

Die Infrastruktur besteht aus einer Monitoring-Komponente, die alle Kernaktivitäten des Case Managers protokolliert. Dazu konsumiert sie Ereignisse, die von anderen Komponenten produziert werden. Je nachdem, ob ein Agenda-Eintrag, ein Artefakt, dessen Beziehung zu einem anderen Artefakt oder der Inhalt eines Artefaktes annotiert wurde, können die Annotationen aus drei Quellen stammen: (1) aus der Agenda-Komponente des adCM-Tools, (2) aus dem Workspace oder (3) aus anwendungsspezifischen Plug-Ins, mit deren Hilfe auch eine einheitliche Benutzeroberfläche für die Erstellung der Annotationen angeboten werden kann. Die Annotationen selbst (nicht die Ereignisse) werden hingegen in einer separaten Serverdatenbank abgelegt.

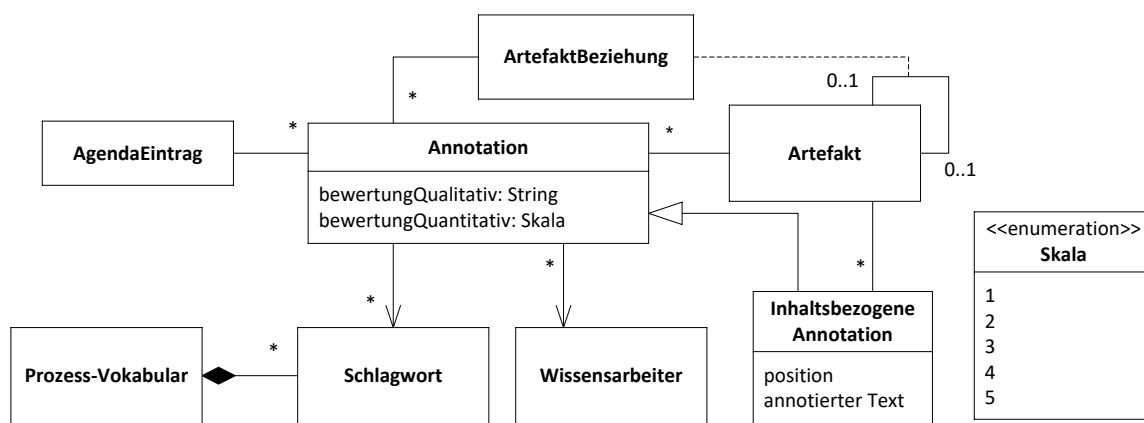


Abbildung 28: Datenmodell für Annotationen

Die Annotationen bestehen jeweils aus einer qualitativen Bewertung in Form eines Kommentars und einer quantitativen Bewertung. Sie verweisen auf Schlagworte des prozessbezogenen Vokabulars und werden von einem Case Manager erstellt. Sofern es sich um eine Annotation auf Inhaltsebene handelt, muss zusätzlich deren Position sowie der annotierte Text innerhalb des

Artefaktes festgehalten werden. Neben der Annotation von Artefakten und deren Inhalt sieht das in Abbildung 7 dargestellte Datenmodell auch die Annotation von Agenda-Einträgen und Beziehungen zwischen Artefakten vor.

Der Ansatz zur Protokollierung von **Planungsvorgängen** ist pragmatisch: Es müssen lediglich die Zustandsveränderungen des aktuellen Falls sowie der Agenda beobachtet und in das zentrale Protokoll, wie in Tabelle 1 gezeigt, serialisiert werden.

5.3.5 ZUSAMMENHANG ZWISCHEN SCHWACH STRUKTURIERTEN PROZESSEN UND EREIGNISPROTOKOLL-KOMPLEXITÄT

Weil traditionelle Discovery-Algorithmen Schwächen in der Anwendung auf komplexe Ereignisprotokolle zeigen (van der Aalst et al. 2005), sollten sie nicht ohne eine vorherige Prüfung des Ereignisprotokolls angewendet werden. Eine solche Prüfung sollte anhand von Metriken Aufschluss darüber geben, ob die Komplexität der Ereignisprotokolle für diese Algorithmen zu hoch ist. Stellen die Metriken eine hohe Komplexität fest, kann dies zwei Ursachen haben:

- a) Es handelt sich zwar nicht um einen komplexen Prozess, aber das Ereignisprotokoll ist fehlerhaft oder unvollständig, weil die Aktivitäten beispielsweise falsch oder nicht zuverlässig protokolliert wurden.
- b) Es handelt sich um einen schwach strukturierten Prozess, dessen unplanbarer Lösungsweg eine hohe Vielfalt an unterschiedlichen Prozesspfaden (Traces) erzeugt. Hier sollten andere Lösungsansätze in Betracht gezogen werden, wie zum Beispiel die in Kapitel 6 beschriebenen Algorithmen zum Mining schwach strukturierter Prozesse.

In diesem Abschnitt soll kurz diskutiert werden, inwiefern schwach strukturierte Prozesse die Komplexität von Ereignisprotokollen beeinflussen. Zu diesem Zweck wird der **Zusammenhang zwischen den besonderen Merkmalen schwach strukturierter Prozesse und der Komplexität von Ereignisprotokollen** hergestellt.

Gemäß der Definition in Kapitel 2.1 sind schwach strukturierte Prozesse in vielerlei Hinsicht unplanbar. Sie zeichnen sich dadurch aus, dass sie viele Varianten besitzen. Diese Varianten unterscheiden sich über weite Strecken in wesentlichen Punkten und fächern sich daher bereits früh im Prozess aus. Weil das Process Mining üblicherweise auf strukturierte Prozesse mit nur wenigen Varianten angewendet wird, interpretieren Process Mining-Forscher eine übermäßige Anzahl an Varianten als **Rauschen** („Noise“), einer nach (van der Aalst 2011) schlechten Qualitätseigenschaft von Ereignisprotokollen. In solchen Fällen wird empfohlen, unerwünschte

Varianten – in (van der Aalst et al. 2005) als „abnormal behavior“ bezeichnet – zu eliminieren. In schwach strukturierten Prozessen ist aber die Vielzahl an Varianten eine Auswirkung der Unplanbarkeit solcher Prozesse. Deshalb kann in Ereignisprotokollen aus schwach strukturierten Prozessen nicht zwischen erwünschten und unerwünschten Varianten unterschieden werden.

Beim Mining strukturierter Geschäftsprozesse ist zu erwarten, dass mit einer zunehmenden Anzahl an protokollierten Instanzen kaum neue Varianten auftreten. Diese Eigenschaft wird als **Vollständigkeit** („Completeness“) bezeichnet. Treten trotzdem immer wieder neue Varianten auf, wird im traditionellen Process Mining schlicht davon ausgegangen, dass das Protokoll unvollständig ist und weiter angereichert werden muss, um ein passendes Prozessmodell daraus erzeugen zu können. Handelt es sich aber um einen schwach strukturierten Prozess, wird das Protokoll wegen der Unplanbarkeit solcher Prozesse prinzipiell immer wieder neue Varianten auftreten, sodass eine Vollständigkeit nur schwierig feststellbar ist. Die fehlende Vollständigkeit und das Rauschen sind also Symptome der Komplexität schwach strukturierter Prozesse, die sich in den Ereignisprotokollen manifestieren.

5.3.6 KOMPLEXITÄTSMETRIKEN FÜR EREIGNISPROTOKOLLE

Um die entworfenen Metriken besser voneinander abgrenzen zu können, wurde eine Klassifikation entwickelt, in die sich die Metriken einsortieren lassen (Benner-Wickner et al. 2014a). Sie unterscheidet zwischen zwei Dimensionen von Komplexität in Ereignisprotokollen: **lokale und globale Komplexität**.

Die **lokale Komplexität** bezieht sich nur auf die Vielfalt, die sich innerhalb jedes einzelnen Traces beobachten lässt. Diese Vielfalt drückt sich unter anderem in der Anzahl unterschiedlicher Ereignisse aus. Ein komplexer Fall enthält demnach viele unterschiedliche Ereignisse, wohingegen ein einfacher Fall folglich nur wenige Ereignisse enthält.

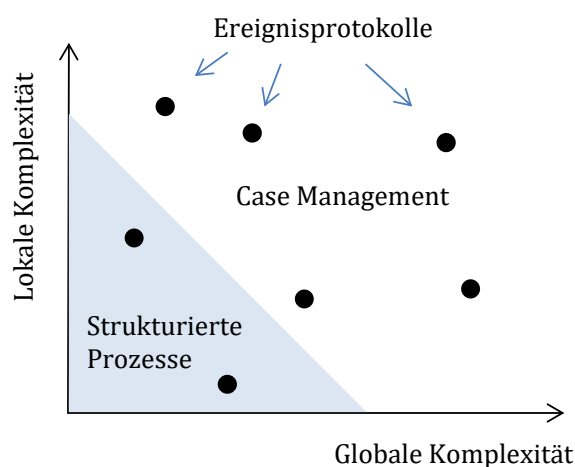


Abbildung 29: Zusammenhang zwischen lokaler/globaler Komplexität und Prozessstypen

Die **globale Komplexität** bezieht sich im Gegensatz zur lokalen Komplexität auf die Vielfalt, die sich zwischen den Traces beobachten lässt. Sie drückt die wechselseitige „Unähnlichkeit“ der Traces untereinander aus. Eine hohe globale Komplexität ist insbesondere dann gegeben, wenn sich die meisten Traces unterscheiden, obwohl man bereits eine hohe Anzahl protokolliert hat. Diese Eigenschaft hängt also stark mit der Vollständigkeit eines Event-Logs zusammen (Yang et al. 2012): Ein schwach strukturierter Prozess wird also – wenn überhaupt – deutlich mehr Instanzen benötigen, um „complete“ zu sein.

Für sich allein genommen deuten weder eine hohe lokale, noch eine hohe globale Komplexität bereits auf einen schwach strukturierten Prozess hin. Viele Prozesse, auch strukturierte, können einen hohen Grad an Komplexität besitzen. Sofern diese Komplexität aber vollständig im Log abgebildet ist, ist sie fassbar/beherrschbar (zum Beispiel durch das Aufteilen von Logs wie bei den Clustering-Verfahren [14]). Zudem können auch Fehler bei der Aggregation von Ereignisdaten eine Ursache dafür sein, dass strukturierte Prozesse ein komplexes Protokoll erzeugen. Ein Domänenexperte kann solche Fehler bei der Analyse der Ereignisprotokolle daran erkennen, dass viele unvollständige Varianten enthalten sind (hohe globale Komplexität), die aber für sich alleine genommen sehr einfach sind (geringe lokale Komplexität).

Im weiteren Verlauf werden Metriken vorgestellt, deren Zweck es ist, die oben beschriebenen Komplexitätsmerkmale aus einem Ereignisprotokoll zu extrahieren. Sie dienen der Entscheidungsunterstützung, ob die vorliegenden Daten

- a) hinreichend strukturiert sind, um die Discovery-Methoden des Process Minings anzuwenden,
- b) einer Vorverarbeitung bedürfen, bevor die Process Mining-Discovery-Methoden angewendet werden können oder
- c) so komplex sind, dass die Lösungsbausteine des adCM angewendet werden sollten.

Es sei vorab angemerkt, dass die vorgestellten Metriken nicht einzeln für sich betrachtet aussagekräftig sind, sondern in ihrer Gesamtheit anzuwenden und zu analysieren sind. Das liegt vor allem daran, dass sie sehr unterschiedliche Aspekte von Komplexität beleuchten. Bei der Vorstellung der Metriken wird mit eher einfachen Formeln zur Messung lokaler Komplexität begonnen, die sich dann aber zu teils umfassenderen Metriken zusammenfügen. Anschließend folgen Metriken zur globalen Komplexität.

Metriken zur Messung lokaler Komplexität

In den Rehabilitationsmanagement-Prozessen werden die versicherten Patienten von der Unfallmeldung bis zu beruflichen Wiedereingliederung von einem Case Manager betreut.

Währenddessen ist der Case Manager über viele Monate hinweg damit beschäftigt, unterschiedliche Therapien zu planen, anzustoßen und zu begleiten, sowie mit einer Vielzahl von Ansprechpartnern sowohl aus dem Gesundheitswesen als auch aus der Versicherungswirtschaft zu zusammenzuarbeiten. Verglichen mit strukturierten Prozessen wie einer Warenbestellung oder einer Reisekostenabrechnung handelt es sich daher um deutlich umfangreichere Arbeiten. Vergleicht man also strukturierte und schwach strukturierte Prozesse, so ist die Größe des Ereignisprotokolls, im Sinne der mittleren Anzahl an Ereignissen je Trace, ein möglicher Indikator für lokale Komplexität. Dies gilt unter der Annahme, dass der Abstraktionsgrad der Ereignisse sich nicht deutlich voneinander unterscheidet.

Definition: Mittlere Trace-Länge (MTL). Sei k die Anzahl an Traces im Ereignisprotokoll L mit $k > 0$. Je Trace mit Index i ($1 \leq i \leq k$) ist n_i die Anzahl an Ereignissen innerhalb des Trace (nicht zu verwechseln mit der Anzahl an unterschiedlichen Ereignissen, die im Nachfolgenden Ereignisklassen genannt werden). Die mittlere Trace-Länge ist wie folgt definiert:

Formel 2: mittlere Trace-Länge

$$MTL(L) = \frac{1}{k} \sum_{i=1}^k n_i$$

Offensichtlich ist diese Metrik ausgesprochen einfach und spiegelt nicht die Struktur des Prozesses wider. Sie unterscheidet auch nicht, ob sich viele Ereignisse wiederholen. In einem solchen Fall wäre der Prozess zwar lang, aber nicht komplex im Sinne vieler unterschiedlicher Ausprägungen. Deshalb wird diese Metrik mit anderen Metriken kombiniert, um präzisere Aussagen über den im Ereignisprotokoll abgebildeten Prozess zu erhalten.

Eine Möglichkeit zur Erweiterung dieser Metrik könnte beispielsweise darin bestehen, die mittlere Anzahl an unterschiedlichen Ereignissen (also Klassen von Ereignissen) innerhalb eines Traces zu zählen. Wenn also zum Beispiel der Rehabilitationsplan im Verlauf des Prozesses mehrfach angepasst wird, so würden diese Ereignisse nicht einzeln, sondern einmalig stellvertretend für die Ereignisklasse gezählt werden. Eine solche Metrik würde im Gegensatz zur mittleren Trace-Länge MTL deutlich robuster gegen Situationen sein, in denen das Protokoll auf den ersten Blick zwar aufgrund seiner Länge sehr komplex erscheint, im Endeffekt aber nur aus sehr vielen, sich wiederholenden Ereignissen besteht. Deshalb wird in diesem Lösungsbaustein eine weitere, im Vergleich zur MTL leicht angepasste Metrik vorgeschlagen, welche die Größe eines Ereignisprotokolls nicht anhand der Ereignisse, sondern anhand der Ereignisklassen misst:

Definition: Normalisierte mittlere Trace-Länge (NMTL). Sei k die Anzahl an Traces im Ereignisprotokoll L . Je Trace mit Index i ($1 \leq i \leq k$) ist E_i die Menge an Ereignisklassen innerhalb des Trace. Sei $|E_i|$ die Mächtigkeit der Menge E_i , dann ist die normalisierte mittlere Trace-Länge definiert als

Formel 3: normalisierte mittlere Trace-Länge

$$NMTL(L) = \frac{1}{k} \sum_{i=1}^k |E_i|$$

Wie bereits oben angedeutet ist es wichtig für die Vergleichbarkeit zweier Ereignisprotokolle, dass ein ähnlicher Abstraktionsgrad für die Erstellung der Ereignisse gewählt wurde. Die NMTL würde beispielsweise schon allein dadurch verdoppeln, dass zu jedem Ereignis auch die sogenannten Lebenszyklusinformationen festgehalten werden, also ob es begonnen oder abgeschlossen wurde.

Um einen möglichst detaillierten Einblick in die lokale Komplexität eines Ereignisprotokolls zu erhalten, wird zusätzlich der Anteil an Wiederholungen genauer untersucht. Denn das Verhältnis von wiederkehrenden Anteilen zur Gesamtlänge eines Prozesses trägt dazu bei, einen Eindruck von der Monotonie zu gewinnen. Zu diesem Zweck wird die folgende Metrik eingeführt:

Definition: Ereignisdichte (ED). Ausgehend davon, dass $MTL(L)$ definiert ist als die mittlere Trace-Länge des Ereignisprotokolls L und dass $NMTL(L)$ definiert ist als die normierte mittlere Trace-Länge, ist die Ereignisdichte des Protokolls wie folgt definiert:

Formel 4: Ereignisdichte

$$ED(L) = \frac{NMTL(L)}{MTL(L)}$$

Diese Metrik setzt die mittlere Anzahl aller Ereignisklassen über alle Traces mit der mittleren Anzahl aller einzelnen Ereignisse ins Verhältnis. Sie beantwortet demnach die Frage, wie hoch der Anteil an wiederkehrenden Ereignissen im Ereignisprotokoll ist. Angenommen die meisten Traces enthalten jedes eingetroffene Ereignis nur einmal, so ist NMTL nur minimal kleiner als MTL. Das Ergebnis der Relation ist in diesem Fall annähernd 1 und beschreibt damit eine sehr hohe Ereignisdichte. Umgekehrt ist die Ereignisdichte gering, wenn aufgrund häufig wiederkehrender Ereignisse NMTL deutlich kleiner ist als MTL. Mithilfe der Ereignisdichte ED kann also ermittelt werden, ob die Größe eines Protokolls von einer Vielzahl unterschiedlicher Ereignisse hervorgerufen wird – und somit definitionsgemäß auf einen schwach strukturierten

Prozess hinweist, oder ob es sich um ein eher monotonen Protokoll handelt, das lediglich aus viel wiederkehrendem Verhalten besteht.

Metriken zur Messung globaler Komplexität

Zur Erinnerung: Im Gegensatz zur lokalen Komplexität, die sich auf die Vielfalt innerhalb eines Traces bezieht, misst die globale Komplexität die Vielfalt, die sich zwischen den Traces beobachten lässt. Um eine solche wechselseitige „Unähnlichkeit“ der Traces untereinander messen zu können, bietet sich die genauere Betrachtung der Wahrscheinlichkeit des Auftretens der Ereignisse an. Angenommen jede im gesamten Protokoll beobachtete Ereignisklasse lässt sich in jedem Trace beobachten. In diesem Fall kann man davon ausgehen, dass sich die Instanzen des Prozesses stark ähneln – schließlich werden immer dieselben Aktivitäten durchgeführt und können sich bestenfalls in ihrer Reihenfolge ändern. Um diesen Sachverhalt auf einfache Weise messbar zu machen, wird NMTL (Anzahl der Ereignisklassen je Trace) ins Verhältnis gesetzt zur Summe aller im Protokoll beobachteten Ereignisklassen:

Definition: Trace-Vielfalt (TV). Sei $NMTL(L)$ die normierte mittlere Trace-Länge zum Ereignisprotokoll L und sei E die Menge der im Protokoll beobachteten Ereignisklassen, dann ist die Trace-Vielfalt wie folgt definiert:

Formel 5: Trace-Vielfalt

$$TV(L) = 1 - \frac{NMTL(L)}{|E|}$$

Nimmt man an, dass jede Ereignisklasse in jedem Trace beobachtet wurde, dann gilt $NMTL(L) \approx |E|$. Die Relation ist demnach 1, was in eine sehr geringe Trace-Vielfalt resultiert (siehe Beispiel in Tabelle 3). Nimmt man das umgekehrte Extrem an, dass nämlich alle im Protokoll beobachteten Ereignisse sich auf alle Traces verteilen, so sind die in NMTL aufsummierten Mengen E_i untereinander disjunkt. In diesem Fall ist die Relation $(1/k)$, was bei einer hinreichend großen Menge beobachteter Trace annähernd 0 ist und somit zu einer hohen Trace-Vielfalt führt.

Tabelle 3: Zwei Extrembeispiele zur Veranschaulichung von TV

L_1		L_2	
i	E_i	i	E_i
1	{a, b, c}	1	{a, b, c}
2	{d, e, f}	2	{a, b, c}

3	{g, h, i}
4	{j, k, l}
5	{m, n, o}
NTML = 3, n = 15	TV(L) = $1-3/15 = 4/5$

3	{a, b, c}
4	{a, b, c}
5	{a, b, c}
NTML = 3, n = 3	TV(L) = $1-3/3 = 0$

Bekanntermaßen enthält ein schwach strukturierter Prozess so viele Varianten, dass sie sich im schlimmsten Fall gar nicht alle mit wirtschaftlich zumutbarem zeitlichen Aufwand in einem Ereignisprotokoll beobachten lassen. Folglich unterscheiden sich die Ereignisprotokolle schwach strukturierter und strukturierter Prozesse dahingehend, dass erstere tendenziell „unvollständiger“ sind. Die oben definierte Metrik ist ein mögliches Maß, um diese Vollständigkeit eines Protokolls beschreiben zu können, denn eine solche Verteilung der Ereignisse lässt sich nur erzeugen, wenn mit jedem Trace Ereignisse hinzukommen, die nicht bereits in anderen Traces beobachtet wurden.

Wie bereits oben angedeutet, berücksichtigt die Metrik nicht die Reihenfolge, in der die Ereignisse auftreten. Diese Information ist aber ebenfalls ein wichtiger Aspekt von Vielfalt. Denn ein Prozess könnte durchaus als komplex wahrgenommen werden, wenn je Fall zwar immer die gleichen Aktivitäten, diese aber stets in einer vollkommen willkürlichen Reihenfolge durchgeführt werden. Es bedarf also einer weiteren Metrik, um diesen Aspekt der Komplexität messen zu können. Erfreulicherweise gibt es in der Forschung bereits gereifte Ansätze, um ähnliche Abschnitte von Traces im Rahmen von Clustering-Verfahren zu identifizieren. Hierbei werden Metriken wie die Hamming- oder die Levenshtein-Distanz verwendet (Levenshtein 1966; Hamming 1950).

Die Hamming-Distanz errechnet die Ähnlichkeit zweier Zeichenketten, in dem die unterschiedlichen Zeichen gezählt werden. Sind die Zeichenketten identisch, ist die Distanz 0. Sind die Zeichenketten komplett verschieden, ist die Distanz gleich der Länge der Zeichenkette (Hamming 1950). Aus dieser Definition wird schon eine Einschränkung der Ähnlichkeitsmetrik erkennbar: Die verglichenen Zeichenketten müssen dieselbe Länge haben. Auf eine solche Einschränkung ist die Definition der Levenshtein-Distanz nicht angewiesen. Sie misst die Ähnlichkeit durch das Zählen von Manipulationsoperationen (Ersetzen, Einfügen und Löschen eines Zeichens), die erforderlich sind zur Transformation der einen Zeichenkette in die andere (Levenshtein 1966). Dementsprechend haben zwei identische Zeichenketten eine Levenshtein-Distanz von 0. Umgekehrt ist die Distanz nach oben begrenzt durch die Länge der längsten Zeichenkette.

In diesem Lösungsbaustein werden solche Verfahren adaptiert und auf die Messung der wechselseitigen Verschiedenheit von Traces angewendet:

Definition: Trace-Disparität (TD). Sei k die Anzahl an Traces und $LD(T_i, T_j)$ die Levenshtein-Distanz zwischen zwei Traces T_i und T_j . Sei außerdem $MTL(L)$ die mittlere Trace-Länge des Protokolls L , dann ist die Trace-Disparität eines Ereignisprotokolls L wie folgt definiert:

Formel 6: Trace-Disparität

$$TD(L) = \frac{2}{k(k-1)MTL(L)} \sum_{i=1}^k \sum_{j=i+1}^k LD(T_i, T_j)$$

Die Entscheidung zwischen der Hamming- und der Levenshtein-Distanz fiel zugunsten von Levenshtein, weil erstere nur die Vertauschungen zwischen Zeichenketten berücksichtigt. Sie ist deshalb nicht in der Lage, unterschiedlich lange Strings zu vergleichen. In der nachfolgenden detaillierten Beschreibung der Funktion LD wird darauf eingegangen, weshalb das eine unbedingt erforderliche Eigenschaft ist.

Die Metrik stellt im Wesentlichen die mittlere Levenshtein-Distanz (LD) dar, normiert gegen die mittlere Länge der Traces MTL . Um diesen mittleren LD -Wert zu erhalten, werden wechselseitig alle LD -Werte zwischen den Traces errechnet, aufsummiert und durch die Anzahl der Summanden ($k(k-1)/2$) geteilt. Die Normierung mit MTL ist notwendig, weil die Levenshtein-Distanz mit der Länge der verglichenen Zeichenketten wächst. Auf diese Weise wird verhindert, dass die Länge eines Ereignisprotokolls, die im Übrigen ja bereits durch die Metrik MTL gemessen wird, die Trace-Disparität beeinflusst.

Weil die Levenshtein-Distanz nur für den Vergleich zweier Zeichenketten definiert ist und nicht für den Vergleich zweier Traces eines Ereignisprotokolls, bedarf es einer Transformation der Eingangsdaten: Es wird eine Abbildung erzeugt, die als Definitionsbereich die Menge an Ereignisklassen besitzt und als Zielbereich einen Unicode-Buchstaben. Wendet man diese Funktion auf alle Ereignisse in einem Trace an, wird der Trace in ein – zugegebenermaßen für Menschen unlesbares – Wort umgewandelt, das wiederum zum Definitionsbereich der Levenshtein-Distanz passt. Dieser Ansatz hat neben seiner Funktion einen wichtigen praktischen Vorteil: Die Länge der Ereignisnamen spielt durch die Abbildung auf genau einen Buchstaben keine Rolle mehr. Allerdings gibt es auch eine Einschränkung, denn die Anzahl an Klassen darf 65.535 ($2^{16}-1$) nicht überschreiten, weil sonst der Speicherbereich für Unicode-Zeichen

überschritten wird. Das bislang umfangreichste Ereignisprotokoll enthielt aber nur 624 Ereignisklassen, sodass diese Obergrenze dem zu lösenden Problem angemessen erscheint.

Die Trace-Disparität kann interpretiert werden als der mittlere Umfang an Änderungen, der zur Umwandlung eines beliebigen Trace in irgendeinen anderen Trace im Ereignisprotokoll notwendig ist. Dieser Zusammenhang lässt sich leicht an einem Beispiel erläutern. Angenommen, ein Ereignisprotokoll enthält fünf Traces mit jeweils vollkommen unterschiedlichen Ereignisklassen (siehe Beispiel 1 in Tabelle 4). Da jedes Ereignis unterschiedlich ist, lautet der mittlere LD-Wert 5. Um von der Länge der Traces zu abstrahieren, normiere man den Wert gegen die mittlere Trace-Länge MTL, die ebenfalls den Wert 5 hat. Das Ergebnis ist eine Trace-Disparität von $5/5 = 1$. Anders ausgedrückt müssen im Schnitt 100% der Ereignisse manipuliert werden, damit sich zwei beliebige Traces aus dem Log gleichen.

Tabelle 4: Zwei Beispiel-Ereignisprotokolle zur Erläuterung der Trace-Disparität

Ereignisprotokoll 1	Ereignisprotokoll 2
<a, b, c, d, e>	<a, b, c, d, e>
<f, g, h, i, j>	<f, g, h, i, j, 1>
<k, l, m, n, o>	<k, l, m, n, o, 2, 3>
<p, q, r, s, t>	<p, q, r, s, t, 4, 5, 6>
<u, v, w, x, y>	<u, v, w, x, y, 7, 8, 9, 0>
$\frac{2}{k(k-1)} \sum_{i=1}^k \sum_{j=i+1}^k LD(T_i, T_j) =$ $\frac{2}{5(5-1)} (20 + 15 + 10 + 5) = \frac{2}{20} 50 = 5$	$\frac{2}{5 \times (5-1)} \sum_{i=1}^k \sum_{j=i+1}^k LD(T_i, T_j) =$ $\frac{2}{5 \times (5-1)} (30 + 24 + 17 + 9) = \frac{2}{20} 80 = 8$
$LD(L) = \frac{5}{MTL(L)} = \frac{5}{5} = 1$	$LD(L) = \frac{8}{MTL(L)} = \frac{8}{7} = 1,142857$ $\frac{8}{ T_{max} } = \frac{8}{9} = 0,8$

Die Metrik zur Messung der Trace-Disparität hat aber auch einen Nachteil, denn sie verlässt unter besonderen Umständen den Wertebereich zwischen 0 und 1. Das liegt daran, dass nicht gegen die *maximale*, sondern gegen die *mittlere* Trace-Länge MTL normiert wird. Dieser Zusammenhang lässt sich anhand des zweiten Beispiels in Tabelle 4 erklären. Das Protokoll hat mit dem ersten Beispiel gemein, dass jeder Trace neue Ereignisse enthält, die Traces also bei einem Vergleich nie gleiche Ereignisse enthalten. Allerdings sind die Traces unterschiedlich lang, weil sie jeweils um unterschiedlich viele, zusätzliche Ereignisse ergänzt wurden (dargestellt durch die Ziffern). Das zweite Beispiel-Ereignisprotokoll hat einen mittleren LD-Wert von $1,142857$ und überschreitet demnach die „Grenze“ von 1. Würde man stattdessen gegen die maximale Trace-Länge $|T_{max}|$

normieren, könnte der Gesamtausdruck nie größer 1 werden. Allerdings ist die Metrik dann schwieriger zu interpretieren und auch weniger robust gegen Ausreißer hinsichtlich der Länge. Denn sobald nur ein Trace in dem Log sehr lang ist, reduziert sich die Metrik stark. Würde man zum Beispiel einen Trace des ersten Protokolls um einen Trace mit 20 unterschiedlichen Ereignissen ersetzen, dann läge die Trace-Disparität nur noch bei 0,55. Umgekehrt ist es beim Normieren gegen MTL nicht vollkommen unbegründet, dass der Wert 1 übersteigen kann. Denn der Anteil des Wertes über 1 zeigt zugleich auf, inwiefern sich die Traces eines Logs nicht nur hinsichtlich der Art der Ereignisse, sondern auch hinsichtlich der Länge unterscheiden. Diese Gründe sprechen dagegen, die maximale Trace-Länge als Referenz für die Normierung einzusetzen.

Adaption von Komplexitätsmetriken aus der Softwaretechnik

Es gibt neben den oben definierten Metriken auch weitere Möglichkeiten, die Komplexität von Geschäftsprozessen anhand ihres Ereignisprotokolls zu messen. Zum Beispiel kann man aus der Softwaretechnik stammende Metriken für die Komplexität des Informationsflusses verwenden (Gruhn und Laue 2006). Hierzu gehören unter anderem Metriken, die das Fan-In und Fan-Out eines Informationsflusses messen (Henry und Kafura 1981). Unter Fan-In versteht man die Anzahl an Startknoten, unter Fan-Out die Anzahl an Endknoten in einem Prozessmodell. In strukturierten Prozessen gibt es oft einen genau definierten Start- und Endpunkt, unter anderem weil der Input, also die zu verarbeitenden Informationen, wohlbekannt ist. Es gibt nur eine übersichtliche Anzahl an Ereignissen, durch die diese Prozesse angestoßen werden und mögliche Ergebnisse sind klar festgelegt. Wenn im Gegensatz dazu aber viele Traces mit jeweils unterschiedlichen Startereignissen beginnen und mit jeweils unterschiedlichen Endereignissen aufhören, handelt es sich entweder um einen ausgesprochen schlecht entworfenen strukturierten Prozess, oder um einen schwach strukturierten Prozess. Denn dann ist es nicht eindeutig, unter welchen Umständen der Prozess gestartet wird und welches Ergebnis er liefert. Dieser Zusammenhang lässt sich leicht am Beispiel des Reha-Managements erläutern, bei dem der sogenannte „Intake“ als erster Schritt des Case Managers nur sehr schwammig definiert ist: Jede mögliche Information, die auf einen Versicherungsfall hindeutet, kann gemäß der deutschen Gesetzeslage (SGB) diesen Prozess anstoßen – von einer handschriftlichen Notiz auf dem Schreibtisch bis hin zu einem formal vollständigen und per Dienstpost eingegangenen Bericht eines Notarztes. Weil es keine klare Definition des Startereignisses gibt und aufgrund der aktuellen Gesetzeslage auch nicht geben wird, handelt es sich gemäß der Metrik also nicht um einen schlecht entworfenen strukturierten Prozess, sondern um einen schwach strukturierten Prozess.

Nachteil einer solchen Metrik ist, dass sie nicht vollkommen unabhängig von der Trace-Disparität ist, denn die darin eingesetzte Levenshtein-Distanz erkennt natürlich auch die vielen unterschiedlichen Start- und Endereignisse. Möchte man aber gezielt die Disparität zu Beginn und am Ende eines Trace untersuchen, ist das Zählen von Fan-In/Fan-Out hingegen vielversprechend. Deshalb wird im nachfolgenden Abschnitt die Rand-Disparität eingeführt.

Definition: Rand-Disparität (RD). Sei n die Gesamtzahl an Ereignisklassen im Log L mit $n > 0$. Je Trace mit Index i ($1 \leq i \leq k$) sei s_i dessen Start- und e_i sein Endereignis. Sei $S := \{s_1, s_2, \dots, s_k\}$ die Menge aller Startereignisse beziehungsweise $E := \{e_1, e_2, \dots, e_k\}$ die Menge aller Endereignisse, dann ist die Rand-Disparität eines Ereignisprotokolls L wie folgt definiert:

$$RD(L) = \frac{(|S| \times |E|)^2}{n}.$$

Das Produkt von Fan-In und Fan-Out entspricht der Anzahl aller möglichen Kombinationen zwischen den Start- und Endereignissen. Henry und Kafura schlagen vor, dieses Produkt per Quadratur gegenüber den anderen Bestandteilen zu verstärken. Damit die Metrik auch beim Vergleich zweier unterschiedlich großer Ereignisprotokolle gut skaliert, wird sie gegen die Anzahl der Ereignisklassen normiert. Nehme man beispielsweise an, zwei unterschiedliche Ereignisprotokolle besitzen jeweils 5 Start- und 5 Endereignisse. Nehme man ferner an, dass das erste Protokoll 20 Ereignisklassen enthält, während das zweite 100 Ereignisklassen enthält. Während bei einer Größe von 100 Ereignisklassen 5 Start- und Endereignisse nicht kurios erscheinen, ist umgekehrt offensichtlich, dass der Prozess mit nur 20 Ereignisklassen und ebenfalls 5 Start- und Endereignissen relativ zu seiner Größe komplexe Schnittstellen zu haben scheint.

Gemäß der obigen Definition berücksichtigt die Rand-Disparität aber nicht die Relevanz der Start- und Endereignisse. Was das genau bedeutet, lässt sich anhand des in Tabelle 5 aufgeführten Beispiels erläutern. In beiden Protokollen wäre die Rand-Disparität identisch ($RD = 9.6$). Es ist aber offensichtlich, dass die meisten Fälle aus dem ersten Protokoll hauptsächlich mit demselben Ereignis a starten und mit m enden. Im zweiten Protokoll hingegen ist die Häufigkeitsverteilung breiter aufgestellt. Es gibt kein dominantes Start- oder Endereignis. Folglich müsste dem zweiten Protokoll prinzipiell eine höhere Komplexität beigemessen werden als dem ersten.

Tabelle 5: Vergleich der Rand-Dispersität zweier unterschiedlicher Ereignisprotokolle (n=15)

Starter-eignisse	Häufigkeit	Ender-eignisse	Häufigkeit	Starter-eignisse	Häufigkeit	Ender-eignisse	Häufigkeit
a	85%	m	85%	a'	21%	m'	32%
b	5%	n	7%	b'	28%	n'	31%
c	5%	o	8%	c'	25%	o'	37%
d	5%			d'	26%		
RD	$(4 \times 3)^2 / 15 = 9.6$			RD	$(4 \times 3)^2 / 15 = 9.6$		

Um dieser Relevanz von Start- und Endereignissen Rechnung zu tragen, muss die Rand-Disparität angepasst werden. Hierzu wird eine Gewichtung eingeführt, die die Häufigkeit der Start- und Endereignisse berücksichtigt. Bezogen auf das Beispiel sollte die Gewichtung dazu führen, dass das erste Protokoll eigentlich nur ein Start- und Endereignis hat, während das zweite Protokoll 4 Start- und 3 Endereignisse hat.

Um das zu erreichen wird eine erweiterte Metrik für die Rand-Disparität eingeführt, die die Standardabweichung der Häufigkeit von Start- und Endereignissen als Gewichtung einbezieht. Ist die Standardabweichung niedrig (d. h. die Auftrittswahrscheinlichkeit aller Start- und Endereignisse ist gleichverteilt), so sollte die Metrik in etwa zu der gleichen Aussage führen wie in der ursprünglichen Rand-Disparität. Ist die Standardabweichung hingegen hoch, sollte die Metrik die Anzahl an Start- und Endereignissen entsprechend heruntergewichten.

Definition: Gewichtete Rand-Disparität (GRD). Sei n die Gesamtzahl an Ereignisklassen im Log L mit $n > 0$. Sei S die Menge aller Startereignisse und E die Menge aller Endereignisse und sei σ_S und σ_E die jeweilige Standardabweichung der Auftrittswahrscheinlichkeit dieser Ereignisse. Dann ist

$|S|' = |S| \times (1 - \sqrt{\sigma_S})$ die gewichtete Anzahl an Startereignissen,

$|E|' = |E| \times (1 - \sqrt{\sigma_E})$ die gewichtete Anzahl an Endereignissen und folglich

$GRD(L) = \frac{(|S|' \times |E|')^2}{n}$ die gewichtete Rand-Disparität eines Ereignisprotokolls L .

Um eine den obigen Anforderungen entsprechend gewichtete Anzahl an Start- und Endereignissen zu erhalten, wurden in verschiedenen Experimenten mit wechselnden

Kombinationen aus Verteilungen und Anzahl an Start- und Endereignissen jeweils unterschiedliche mathematische Operatoren eingesetzt. Im Ergebnis hat sich herausgestellt, dass der Faktor $(1 - \sqrt{\sigma})$ dem erwarteten Ergebnis am nächsten kommt. Wendet man die gewichtete Rand-Disparität beispielsweise auf das obige Beispiel an, so erhält man als gewichtete Anzahl an Startereignissen für das erste Protokoll einen Wert von 1,23 und für das zweite Protokoll einen Wert von 3,8. Diese Werte treffen die Erwartungen (1 und 4) also ziemlich gut. Aufgrund der angepassten Anzahl an Start- und Endereignissen erzielt man mit der gewichteten Rand-Disparität für das erste Protokoll einen Wert von 0,12 und für das zweite Protokoll einen Wert von 7,87. Beide Werte unterscheiden sich deutlich von dem ursprünglichen, ungewichteten Wert von 9.6 und berücksichtigen die offensichtlich unterschiedliche Komplexität beider Protokolle.

5.4 EVALUATION DER PROTOKOLLIERUNG

Dieser Abschnitt soll zeigen, ob sich die im Lösungskonzept aufgeführten Bausteine zur Protokollierung von schwach strukturierten Prozessen an einem Fallbeispiel umsetzen lassen. Inwiefern die Bausteine für die nachfolgenden Algorithmen wie zum Beispiel dem Template Mining nützlich sind, ist Gegenstand der jeweiligen Evaluationsschritte der nachfolgenden Bausteine.

Gemäß der Anforderungen aus Kapitel 5.1.2 ist zu zeigen, ob es ein Werkzeug gibt, das neben der Arbeit mit der Agenda und den Artefakten auch weitere Kernaktivitäten eines Case Managers protokolliert (Anforderung 9), und zwar verlässlich, vollständig, in einer wohldefinierten Struktur und mit einer eindeutigen Semantik (Anforderung 10).

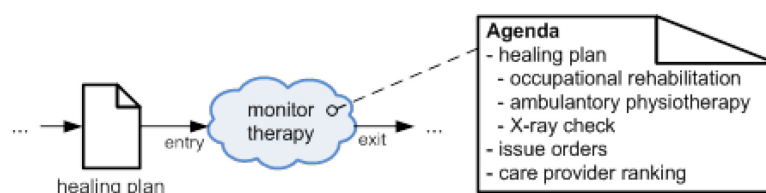


Abbildung 30: Ein Fallbeispiel aus dem Reha-Management

5.4.1 EINFÜHRUNG IN DAS FALLBEISPIEL

Als Fallbeispiel wird das durchgehende Beispiel aus dem Rehabilitationsmanagement verwendet (siehe Abbildung 30). Der Ablauf der Interaktion wird durch den folgenden „Happy Path“-Testfall definiert, der die Grundfunktionalität der Protokollierung abdecken soll:

Herr Müller, Reha Manager bei der Berufsgenossenschaft für Nahrungsmittel und Gastgewerbe, erhält die Nachricht von seinem Teamleiter, dass ihm ein neuer Fall zugewiesen wurde: Herr Schmidt hat sich durch einen Sturz auf der Arbeit einen Oberschenkelhalsbruch zugezogen und muss während der Therapie begleitet werden. Herr Müller beginnt die Bearbeitung des Falls, indem er das adCM-Werkzeug öffnet und sich authentifiziert (Schritt 1). Anschließend beginnt er einen neuen Fall mit dem Namen „Reha Herr Schmidt“ (2) und fügt mit „Heilverlaufsplan“ einen grundlegenden Agendaeintrag hinzu (3). Aus Erfahrung weiß Herr Müller, dass bei solchen Fällen umfangreiche Röntgenaufnahmen anfallen und fügt auch schon mal den Eintrag „Röntgenaufnahmen“ hinzu. Aus dem digitalen Posteingang des Dokumentenmanagementsystems entnimmt er das Diagnosedokument, die Medikationsinformationen sowie den zuvor mit Patient und Arzt vereinbarten Heilverlaufsplan und ordnet die Dokumente als Artefakte dem Agendaeintrag „Heilverlaufsplan“ zu (4). Um deutlich zu machen, dass der Heilverlaufsplan auf Grundlage der Diagnose erstellt wurde, platziert er beide Artefakte direkt nebeneinander auf den Workspace und verbindet sie mit der Semantik „basiert auf“ (5). Anschließend öffnet er den Heilverlaufsplan, informiert sich über die wesentlichen Therapieschritte (6) und will sie als Agendaeinträge dem Eintrag „Heilverlaufsplan“ hinzufügen. Dabei fällt ihm auf, dass dort bereits ein Schritt für die Röntgenaufnahmen vorgesehen ist. Um diesen Eintrag nicht doppelt zu führen, sortiert er den alten Agendaeintrag in den Heilverlaufsplan ein und ergänzt die anderen Schritte (7). Bis die ersten Rückmeldungen aus den Reha-Maßnahmen erfolgen, macht sich Herr Müller als eine Art Lesezeichen eine kleine Notiz in den aktuellen Agendaeintrag, pausiert den Fall und widmet sich anderen Versicherten (8).

Nach zwei Wochen Reha tritt eine Komplikation bei Herrn Schmidt durch Medikamentennebenwirkungen auf (Gewichtsverlust), die eine Neuausrichtung des Heilverlaufsplans erfordert. Herr Müller öffnet den Fall wieder und sucht in dem Diagnosedokument nach der betroffenen Indikation (9). Anschließend kennzeichnet er die Reha als abgeschlossen, passt den Heilverlaufsplan an die neuen Entwicklungen an und aktualisiert gemäß den Dokumentationsvorschriften das im Dateinamen enthaltene Datum (10). Außerdem annotiert er das veraltete Medikationsdokument mit einem Hinweis über die aufgetretene Komplikation und ersetzt es durch ein neues Dokument (11). Die Neuausrichtung des Heilverlaufsplans sieht eine langfristige Medikamentenumstellung vor, die die ambulante Physiotherapie vorerst ersetzt (12).

Nach einigen Monaten komplizierter Therapiemaßnahmen, in deren Verlauf Herr Müller gelegentlich Maßnahmen wieder aufgreifen, umsortieren oder umbenennen musste (13), sind alle Einträge nunmehr erledigt (14). Herr Müller reflektiert den Fall in einer Post-Mortem-

Besprechung mit seinem Teamleiter. Beide kommen zum Schluss, dass die Wiedereingliederung von Herrn Schmidt zwar erfolgreich war, Herr Müller aber noch den Abschlussbericht ergänzen muss, um den Fall zu den Akten legen zu können (15).

Um zu zeigen, ob das oben beschriebene Szenario alle Ereignisse aus dem Datenmodell und damit alle Kernaktivitäten abdeckt, werden in Tabelle 6 die Schritte den Anforderungen zugeordnet.

Tabelle 6: Abdeckung der Ereignisse durch den Testfall

Schritt	Fall				Agenda			Agendaeintrag							Artefakt								
	gestartet	unterbrochen	fortgeführt	abgeschlossen	erledigt	unerledigt	umsortiert	selektiert	erledigt	unerledigt	hinzugefügt	gelöscht	umbenannt	annotiert	manipuliert	umsortiert	geöffnet	durchsucht	hinzugefügt	gelöscht	umbenannt	annotiert	verknüpft
1																							
2	X																						
3								X			X												
4																			X				
5																X							X
6																	X						
7							X	X			X												
8		X												X									
9			X														X	X					
10									X						X		X				X		
11																			X	X		X	
12											X	X											
13										X			X										
14					X																		
15				X	X	X													X				

Die in Kapitel 5.3.1 beschriebenen Lösungskonzepte wurden zur Evaluation mit einem Prototyp implementiert. Anschließend wurde das oben beschriebene Szenario in Form eines automatisierten Akzeptanztests nachgestellt. Hierzu wurde das Werkzeug SikuliX (<http://sikulix.com>) benutzt. Der erstellte Testfall ist in Anhang 10.1 zu finden.

5.4.2 EVALUATION DER VOLLSTÄNDIGKEIT DER PROTOKOLLIERUNG

Um die Anforderung an die Vollständigkeit des Ereignisprotokolls zu prüfen, erfolgte ein Abgleich des erzeugten Ereignisprotokolls (siehe Anhang) mit den Ereignisklassen. Das Ergebnis wird in der nachfolgenden Tabelle dokumentiert:

Tabelle 7: Überprüfung, ob die Ereignisse vollständig protokolliert wurden

	Fall				Agenda			Agendaeintrag							Artefakt								
	gestartet	unterbrochen	fortgeführt	abgeschlossen	erledigt	unerledigt	umsortiert	selektiert	erledigt	unerledigt	hinzugefügt	gelöscht	umbenannt	annotiert	manipuliert	umsortiert	geöffnet	durchsucht	hinzugefügt	gelöscht	umbenannt	annotiert	verknüpft
OK ?	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X

Der Abgleich zeigt, dass alle Benutzerinteraktionen vollständig dokumentiert wurden.

5.4.3 EVALUATION DER VERLÄSSLICHKEIT DER PROTOKOLLIERUNG

Um die **verlässliche** Protokollierung zu überprüfen, wurden die Tests mithilfe des Werkzeugs für automatisierte Benutzerakzeptanztests mehrfach wiederholt. Dabei wurden zwei unterschiedliche Systeme (Mac OS-System und Windows-System) zu unterschiedlichen Zeitpunkten verwendet. Die erzeugten Protokolldateien haben aber jederzeit die im Anhang enthaltene Protokolldatei reproduziert – unter Ausnahme der unterschiedlichen Zeitstempel.

5.4.4 EVALUATION DER WOHLDEFINIERTHEIT DER PROTOKOLLSTRUKTUR

Ob die Struktur des erzeugten Protokolls tatsächlich **wohldefiniert** ist, muss in vielfältiger Hinsicht geprüft werden. Erstens muss es sich um eine *wohlgeformte* XML-Datei handeln. Das heißt, die XML-Datei muss eine korrekte XML-Syntax aufweisen. Diese Prüfung wurde mit dem offiziellen XML-Validator der Organisation W3C durchgeführt (<https://validator.w3.org/check>). Im Ergebnisbericht wurde die Wohlgeformtheit der XML-Syntax bestätigt mit dem Hinweis, dass die Angabe der verwendeten Zeichenkodierung hinzugefügt werden sollte.

Ein weiterer Aspekt der Prüfung ist, ob das Protokoll konform zum XES-Standard ist. Die Entwickler des Standards bieten dafür eine XML-Schemadefinition (XSD) an, gegen die validiert werden kann (<http://www.xes-standard.org/xes.xsd>). Hierzu muss die Schemadefinition im Kopf der XML-Datei referenziert werden (siehe Listing 2).

```
<log xes.version="1.4" xes.features="nested-attributes" openxes.version="1.8"
xmlns="http://www.xes-standard.org/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.xes-standard.org/xes.xsd">
```

Listing 2: Angabe der XML-Schemadefinition im Ereignisprotokoll

Anschließend wurde durch ein Plug-In des Werkzeugs Notepad++ (<https://notepad-plus-plus.org/>) geprüft, ob die XML-Datei valide ist, beziehungsweise zu dem referenzierten Schema passt. Das Ergebnis war zunächst negativ, unter anderem weil die Zeitstempel in den Ereignissen nicht mit dem Datentyp `xs:dateTime` übereinstimmten. Anstatt das Datum gemäß der XML-Spezifikation mit dem Buchstaben „T“ von der Uhrzeit zu trennen, wurde ein einfaches Leerzeichen hierfür verwendet. Nachdem dieser Fehler in der Ereigniserzeugung behoben wurde und die Groß-/Kleinschreibung der Boolean-Werte korrigiert wurde, ist die Prüfung gegen das XML-Schema des XES-Standards erfolgreich verlaufen.

5.4.5 EVALUATION DER PROTOKOLLIERUNG SEMANTISCH VERKNÜPFTER EREIGNISSE

Um das Lösungskonzept aus Kapitel 5.3.2 zur Protokollierung semantisch verknüpfter Ereignisse zu evaluieren, wurde eine solche Ontologie implementiert und auf einem Server bereitgestellt. Die korrekte Funktionalität der Ontologie wurde mithilfe einer SPARQL-Abfrage überprüft. Die Abfrage gibt die semantisch angereicherten Protokolldaten aus dem Beispielszenario zurück, welche mit dem XML-Ereignisprotokoll abgeglichen wurden (siehe Anhang 10.2). Das Listing 3 zeigt diese Abfrage am Beispiel des Falls mit der ID 52f9fa2e-28be-4114-90ed-ac650721897c und wird im nächsten Abschnitt kurz erläutert.

```
PREFIX adcm: http://paluno.uni-due.de/adcm#
PREFIX case: http://paluno.uni-due.de/opc/example/

describe ?events
WHERE {
  {
    case:52f9fa2e-28be-4114-90ed-ac650721897c\ / adcm:caseEventTrace ?eventTrace.
    ?eventTrace adcm:containsEvent ?events.
  }
  UNION {
    case:52f9fa2e-28be-4114-90ed-ac650721897c\ / adcm:caseEventTrace ?eventTrace.
    ?eventTrace adcm:containsEvent ?firstEvent.
    ?firstEvent adcm:nextEvent+ ?events
  }
}
```

Listing 3: SPARQL-Abfrage zur Ausgabe der in der Ontologie gespeicherten Protokolldaten

Mit dem SPARQL-Befehl „describe“ werden alle mit den Ereignissen unmittelbar verknüpften Informationen ausgegeben. Wie die Ereignisse aus der Datenbank abgefragt werden können, ist Gegenstand des „WHERE“-Teils der Abfrage. Weil die Ereignisse nicht unmittelbar dem Fall zugeordnet sind, sondern dessen Trace, muss dieser zunächst im ersten Segment der WHERE-Klausel selektiert werden. Ab dann ist der Trace der SPARQL-Variablen „?eventTrace“ zugewiesen, die dann in der darauffolgenden Zeile verwendet werden kann, um alle Ereignisse dieses Traces zu selektieren. Beließe man die WHERE-Klausel in diesem Zustand, erhielte man

aber nicht die Liste aller Ereignisse, sondern nur genau das erste Ereignis (zur Erinnerung: weil graphbasierte Datenbanken keine Listenstrukturen verwenden, sind die Ereignisse als verkettete Liste abgelegt). Deshalb wird das zweite Segment in der WHERE-Klausel benötigt, dessen Selektion mit der ersten über das UNION-Schlüsselwort zusammengefügt wird. Das zweite Segment selektiert nach dem gleichen Prinzip zunächst das erste Ereignis und legt es unter der Variablen „?firstevent“ ab. In der letzten Zeile des Segments wird dann das erste Ereignis verwendet, um alle transitiv verbundenen Ereignisse zu selektieren (Abfragen über transitive Beziehungen werden in SPARQL mit dem „+“-Operator realisiert).

Tabelle 8: Ergebnis des Abgleichs zwischen dem dateibasierten und dem semantischen Ereignisprotokoll

	Fall				Agenda			Agendaeintrag							Artefakt								
Kontext #	gestartet	unterbrochen	fortgeführt	abgeschlossen	erledigt	unerledigt	umsortiert	selektiert	erledigt	unerledigt	hinzugefügt	gelöscht	umbenannt	annotiert	manipuliert	umsortiert	geöffnet	durchsucht	hinzugefügt	gelöscht	umbenannt	annotiert	verknüpft
1	X	X	X	X	X	X	X	X	X	X	X	X	X	X			X		X	X	X	X	X

Ob das semantisch verknüpfte Ereignisprotokoll konsistent ist mit dem dateibasierten XES-Ereignisprotokoll, zeigt der Abgleich zwischen beiden Protokollen in der Tabelle 8. Demnach werden die meisten der Ereignisse auch in der Ontologie gespeichert. Die fehlenden drei Ereignistypen werden zwar vom Client erzeugt, lokal in der XES-Protokolldatei gespeichert und auch an die Ontologie gesendet. Aber die Implementierungsarbeiten zur Verarbeitung der Ereignisse in der Ontologie wurden aus Zeitmangel nicht fertiggestellt.

5.5 EVALUATION DER KONTEXTERMITTLUNG

Um zu überprüfen, ob der Lösungsansatz zur Ermittlung des jeweiligen Fallkontextes im Zusammenspiel mit den anderen adCM-Konzepten funktioniert, wurden zwei unterschiedliche Testszenarien erarbeitet. Im ersten Szenario wurden vier künstlich erzeugte Fälle entlang des bekannten Reha Management-Fallbeispiels verwendet, deren Artefakte mit charakteristischen Texten beziehungsweise Stichworten versehen wurden. Im zweiten Szenario werden ebenfalls vier Fälle verwendet. Sie werden allerdings nicht künstlich erzeugt, sondern aus einer Datenbasis der Landesärztekammer Rheinland-Pfalz entnommen. Es handelt sich dabei um eine öffentliche

Fallsammlung des Schlichtungsausschusses zu ausgewählten Fachgebieten mit textuellen Fallbeschreibungen und den zugehörigen Gutachten².

5.5.1 AUSWAHL DER FALLBEISPIELE – SZENARIO 1

Der erste Fall (1A) ist das bekannte Fallbeispiel. Mithilfe der Kontexttermvektoren soll die Ähnlichkeit der anderen drei Fälle (1B-1D) zu diesem Fall gemessen werden. Der zweite Fall (1B) ist eine minimal angepasste Variante des Fallbeispiels (siehe Abbildung 31). Er enthält beispielsweise die gleiche Diagnose, die nur leicht verschieden aufgeschrieben wurde. Mit ihm soll gezeigt werden, ob diese minimalen Abweichungen auch nur minimal in den Vergleichswerten abgebildet werden. Der Fall 1C hingegen ist so abgewandelt worden, dass eine weitere Diagnose enthalten ist sowie eine besondere Medikation, die aufgrund eines Leberschadens verordnet wurde. Der Heilverlaufsplan von 1C enthält ebenfalls zusätzliche Elemente, um die Unterschiede bei der Diagnose zu berücksichtigen. Der vierte Fall (1D) hat keinerlei inhaltliche Übereinstimmung mit den anderen Fällen. Es handelt sich um eine Verbrennungsverletzung, die gänzlich anders behandelt wird.

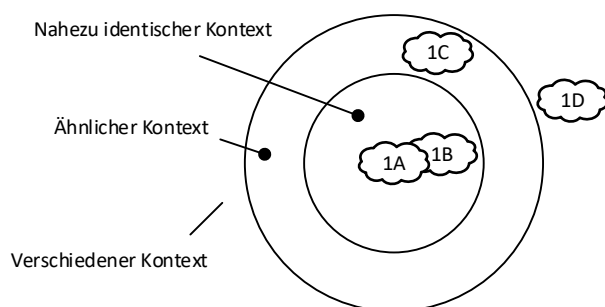


Abbildung 31: Ähnlichkeit der Fälle 1B-1D zu 1A

Inwiefern das vorliegende Konzept zur Kontextidentifizierung dazu herangezogen werden kann, diese Fälle entsprechend ihrer hypothetischen Ähnlichkeit zu unterscheiden, wird in Kapitel 0 diskutiert.

5.5.2 AUSWAHL DER FALLBEISPIELE – SZENARIO 2

Die Fälle aus der Datensammlung wurden nach dem ähnlichen Prinzip ausgewählt wie in Abbildung 31 dargestellt. Der erste Fall (2A) enthält dieselbe Diagnose wie das durchgehende Fallbeispiel, es handelt sich also ebenfalls um eine Fraktur des Oberschenkelhalses. Der zweite Fall (2B) ist nur insofern eine möglichst minimale Abweichung von 2A, als dass der Fall zu

² <http://www.laek-rlp.de/aerzteservice/fortbildung/fallbeispiele/index.php> (zuletzt aufgerufen am 12.07.2016)

demselben Fachgebiet gehört und dieselbe Diagnose vorliegt. Trotzdem sind die Patienten und der Behandlungspfad natürlich sehr unterschiedlich. Die Datenbasis ist nicht groß genug, um Fälle mit einer stärkeren Ähnlichkeit auswählen zu können. Der dritte Fall (2C) soll einen ähnlichen Fall repräsentieren, indem zwar das gleiche Fachgebiet, aber eine andere Art des Knochenbruchs (Multiple Mittelhandfraktur) ausgewählt wurde. Der letzte Fall (2D) soll eine möglichst starke Abweichung des Kontextes aufweisen. Hierzu wurde ein gänzlich anderes Fachgebiet gewählt (Gynäkologie statt Unfallchirurgie).

Ob auch bei solchen Praxisbeispielen die Kontextidentifizierung dazu herangezogen werden kann, die Ähnlichkeiten korrekt abzubilden, wird in Kapitel 5.5.4 diskutiert.

5.5.3 ERGEBNISSE – SZENARIO 1

Für jeden der synthetisch erzeugten Fallbeispiele wurde durch die Kontextidentifizierungskomponente ein Kontexttermvektor erzeugt. Auszüge aus diesen Vektoren sind im Anhang 10.4 dokumentiert. Anschließend wurden die Kontextvektoren auf Grundlage der Kosinus-Ähnlichkeit wechselseitig miteinander verglichen.

Definition: Kosinus-Ähnlichkeit

Die Kosinus-Ähnlichkeit ist ein Maß für die Ähnlichkeit zweier Vektoren. Sie wird berechnet über den Kosinus des Winkels, der zwischen den beiden Vektoren aufgespannt wird. Im Vektorraummodell sind demnach zwei Vektoren identisch, wenn der Winkel gleich 1 ist. Das ist genau dann der Fall, wenn sie dieselben Terme enthalten. Zwei Vektoren sind komplett verschieden, wenn der Winkel gleich 0 ist, die Vektoren also orthogonal zueinanderstehen. Das ist genau dann der Fall, wenn die Terme keine Schnittmenge bilden. (Weil im Vektorraummodell die Elemente der Vektoren je nach Zählweise die Existenz oder Häufigkeit von Termen beschreiben, sind sie nie negativ. Dadurch können Vektoren nicht entgegengerichtet sein, wodurch wiederum der Kosinus des Winkels nie negativ ist.)

Die wechselseitige Berechnung der Kosinus-Ähnlichkeit zwischen den Dokumenten der Fallbeispiele zeigt die Tabelle 9.

Tabelle 9: Matrix mit den Kosinus-Ähnlichkeiten zwischen den Fällen 1A-1D

	1A	1B	1C	1D
1A	1,0000	0,9487	0,8468	0,5531

1B	-	1,0000	0,8197	0,5609
1C	-	-	1,0000	0,5802
1D	-	-	-	1,0000

Die Kontexte zwischen Fall 1A und 1B werden erwartungskonform als beinahe identisch eingestuft. Die Ähnlichkeit der Kontexte zwischen 1A und 1C fällt geringer aus und reflektiert so auf korrekte Weise die Annahme, dass es sich nicht um einen identischen, aber zumindest um einen vergleichbaren Fall handelt. Die Ergebnisse des Kontextvergleichs zwischen den Fällen 1A und 1D zeigen, dass sie sich im Vergleich zu den anderen beiden Fällen deutlich weniger ähnlich sind. Trotzdem ist dieses Teilergebnis nicht vollständig zufriedenstellend, weil es sich eigentlich um einen gänzlich anderen Fall handeln sollte und die Übereinstimmung des Kontextes, naiv betrachtet, gegen null gehen sollte. Dass das nicht eingetreten ist, lässt sich aber dadurch erklären, dass trotz der unterschiedlichen Diagnosen in beiden Fällen Paracetamol verschrieben wurde, aufgrund der längeren stationären Behandlung eine Physiotherapie notwendig war und nicht zuletzt natürlich in beiden Fällen ein Heilverlaufsplan benötigt wurde.

Zur Überprüfung der Ergebnisse wurden die Kontexte auch ausgehend von 1B und 1C verglichen. Erwartungsgemäß ist die Ähnlichkeit zwischen 1B und 1C vergleichbar mit der Ähnlichkeit zwischen 1A und 1C, schließlich sind 1A und 1B beinahe identisch. Auch die Ähnlichkeiten zu 1D sind jeweils vergleichbar.

Abschließend lässt sich festhalten, dass auf Grundlage der synthetischen Fallbeispiele die Kontextidentifikation wie erwartet funktioniert. Es besteht Grund zu der Annahme, dass nachfolgende Lösungsbausteine die im Monitoring kontinuierlich beobachtete Ähnlichkeit zu anderen Fällen verwenden können, um zum Beispiel hilfreiche Artefakte oder Agendaeinträge aus diesen Fällen vorzuschlagen. Um diese Annahme weiter zu bekräftigen werden im nächsten Abschnitt die Ergebnisse zu den Fallbeispielen aus der Praxis diskutiert.

5.5.4 ERGEBNISSE – SZENARIO 2

Neben den synthetisch erzeugten Fallbeispielen wurden auch die Beispielfälle aus der Praxis analysiert, um Kontexttermvektoren für die Artefakte zu erzeugen. Auszüge aus diesen Vektoren sind ebenfalls im Anhang 10.4 dokumentiert. Die Kosinusähnlichkeiten werden in der Tabelle 10 aufgelistet.

Tabelle 10: Matrix mit den Kosinus-Ähnlichkeiten zwischen den Fällen 2A-2D

	2A	2B	2C	2D
2A	1,0000	0,3995	0,3794	0,2672
2B	-	1,0000	0,4530	0,3079
2C	-	-	1,0000	0,1596
2D	-	-	-	1,0000

Beginnt man bei der Analyse mit den Ähnlichkeiten zu Fall 2A so fällt auf, dass der zweite Fall im Vergleich zu den synthetischen Fällen keine so hohe Übereinstimmung mit dem ersten Fall aufweist. Doch wie kurz bei der Fallauswahl beschrieben liegt die Ursache hauptsächlich darin, dass zwar die Diagnose und das Fachgebiet gleich sind, jedoch der ausführlich dokumentierte Behandlungsverlauf trotzdem noch sehr individuell ist. Darüber hinaus sind alle anderen Ähnlichkeiten im Sinne einer Ordinalskala erwartungskonform ($2B > 2C > 2D$). Die Unterschiede sind aber nicht so gravierend, wie es in einem perfekt konstruierten Beispiel möglich ist. Trotzdem sind die Differenzen zu dem gänzlich anderen Fallbeispiel 2D deutlich erkennbar.

Die Ergebnisse ausgehend von 2B und 2C betrachtet unterliegen zwar stärkeren Schwankungen, sind aber trotzdem bezogen auf die Ordinalität korrekt. Insgesamt setzt sich das positive Ergebnis aus den synthetischen Testfällen in den Praxisbeispielen fort. Die Kontexte können korrekt ermittelt und verglichen werden, um durch nachfolgende Lösungsbausteine nützliche Artefakte oder Agenda-Einträge vorzuschlagen.

5.6 EVALUATION DER KOMPLEXITÄTSMETRIKEN

Um die Validität der Metriken evaluieren zu können, also um zu zeigen, ob sie tatsächlich die in Kapitel 5.3.6 beschriebenen Komplexitätsaspekte messen können, wurde ein zweistufiges Evaluationsdesign konzipiert und durchgeführt (Benner-Wickner et al. 2014a). Im ersten Schritt wurden Experimente mit künstlich erzeugten Ereignisprotokollen durchgeführt, mit denen sich die einzelnen Komplexitätsaspekte an Extrembeispielen gut zeigen und erklären lassen. Um die Praxistauglichkeit der Metriken zu erforschen, wurden im zweiten Schritt aber auch echte Ereignisprotokolle aus der Praxis eingesetzt und im Sinne einer Fallstudie mit den Metriken bewertet. Im ersten Abschnitt dieses Unterkapitels werden das Design und die Ergebnisse aus den Experimenten mit künstlichen Ereignisprotokollen beschrieben. Der zweite Abschnitt diskutiert das Design und die Ergebnisse aus den Fallstudien. Zusammengefasst soll damit gezeigt werden, ob die Metriken Hinweise auf eine unzureichende Ereignisprotokollqualität (hier im

Sinne von zu hoher Komplexität) für traditionelles Process Mining liefern können. Ein solcher Hinweis würde dann bedeuten, dass dedizierte Case Management-Techniken wie adCM angewendet werden sollten.

5.6.1 EXPERIMENT MIT SYNTHETISCHEN LOG-DATEIEN

Für das Experiment wurden vier künstliche Prozesse modelliert und anschließend mithilfe von speziellen Werkzeugen simuliert, um entsprechende Ereignisprotokolle zu erhalten. Beim Design der Prozesse wurde vor allem darauf geachtet, dass die erzeugten Protokolle aus der Simulation möglichst genau diejenigen besonderen Merkmale von Komplexität aufweisen, die von den Metriken identifiziert werden sollen. Die auf diese Weise produzierten Daten belaufen sich auf 6000 Traces mit insgesamt 54.242 Ereignissen. Als Werkzeuge kamen dabei das PLG Framework (Burattin und Sperduti 2011) sowie die CPN Tools (Ratzer et al. 2003) zum Einsatz. Tabelle 11 zeigt die Eckdaten der erzeugten Protokolle.

Tabelle 11: Details zu den synthetischen Beispielprotokollen

ID	Beschreibung	Traces	Ereignisse	Ereignisklassen
L ₁	Lange Sequenzen, keine Zyklen	1000	13 000	14
L ₂	Lang, aber aufgrund von Zyklen	1000	13 034	9
L ₃	Kurz und variantenreich	1000	6 166	38
L ₄	Lang und komplex	1000	11 042	20
L ₅	Kurz, aber hohes Fan-in/out	1000	4 000	12
L ₆	Kurz, mit ungleich verteiltem Fan-in/out	1000	4 000	12

An dieser Stelle sei angemerkt, dass die Metriken nicht in der Lage sind, das in der Process Mining-Community als „Rauschen“ (engl. noise) bezeichnete Verhalten explizit messbar zu machen. Dabei handelt es sich um seltenes und deshalb im Process Mining eher unerwünschtes Verhalten, das in einem Ziel-Prozessmodell zwecks besserer Lesbarkeit nicht auftreten soll. Dass die Metriken dieses Rauschen quasi ignorieren ist aber kein Nachteil, sondern eher umgekehrt: Weil es aufgrund der hohen Diversität eben gerade nicht den einen „Happy Path“ durch einen schwach strukturierten Prozess gibt, ist kaum ein Unterschied zwischen seltenem und häufigem Verhalten messbar. Aus diesem Grund wurden die Protokolle auch ohne ein solch künstlich erzeugtes Rauschen generiert.

Im weiteren Verlauf wird die Struktur der künstlich erzeugten Prozesse einzeln diskutiert und es wird gezeigt, welche Merkmale von Prozesskomplexität sie adressieren. Das in Abbildung 32 gezeigte Prozessmodell zu Ereignisprotokoll L₁ beispielsweise ist so beschaffen, dass es als Stellvertreter für lange, aber einfache Prozesse stehen soll. Mit dessen Hilfe soll untersucht

werden, wie sich die Metriken bei Ereignisprotokollen verhalten, die kaum alternative Prozesspfade, stattdessen aber vergleichsweise ausgeprägte Sequenzen aus unterschiedlichen Ereignissen enthalten. Es ist zu erwarten, dass die Protokolle eine hohe lokale Komplexität aufweisen, weil die Ereignisdichte aufgrund der fehlenden Zyklen sehr hoch ist.

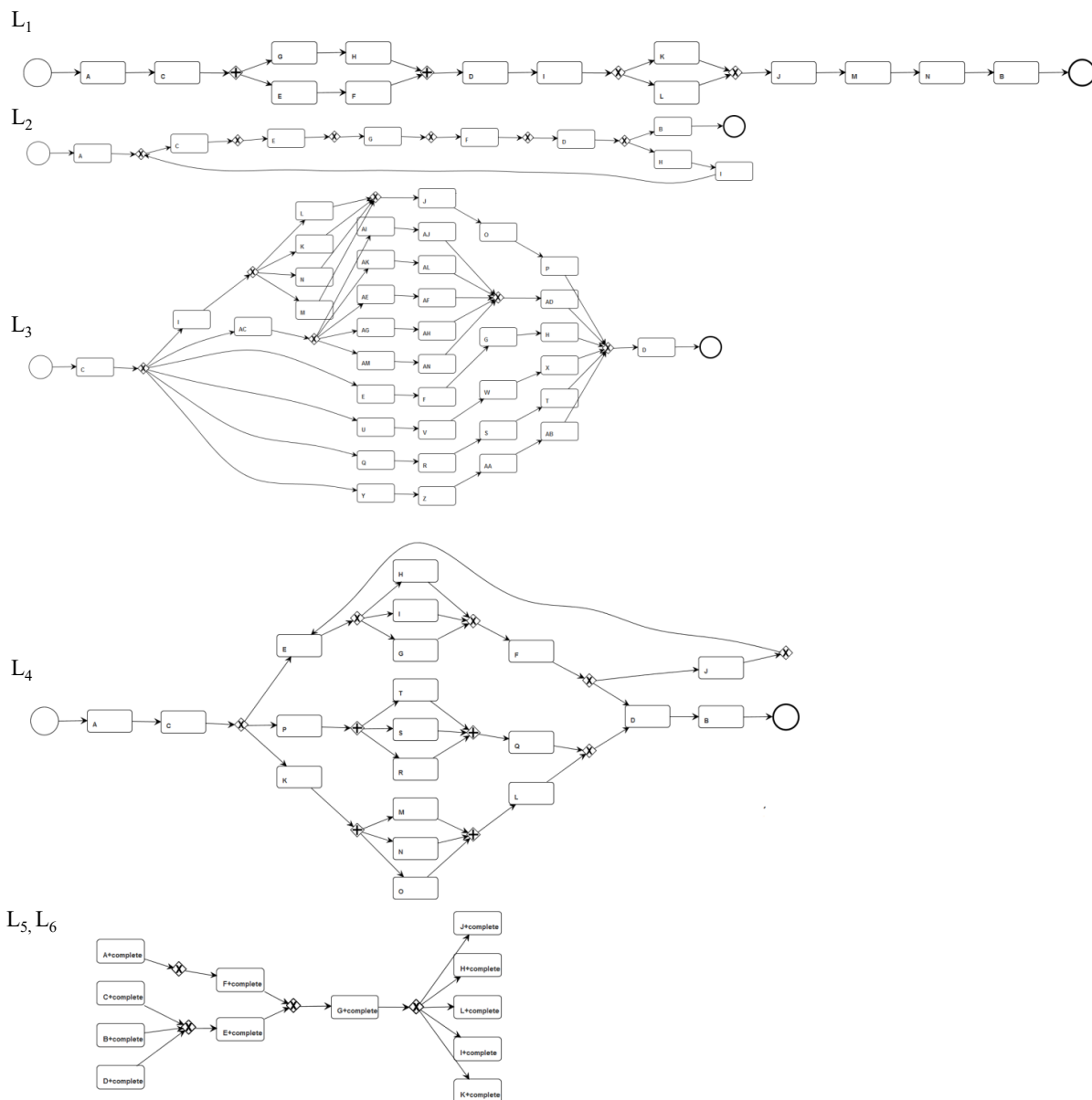


Abbildung 32: Modelle der künstlichen Prozesse, die zur Erzeugung der Ereignisprotokolle L1 (oben) bis L5 (unten) herangezogen wurden

Der Prozess zu L₂ wurde erzeugt um zu überprüfen, inwiefern die Metriken einen Hinweis darauf geben können, ob die Länge eines Traces maßgeblich aus wiederkehrendem Verhalten resultiert. Das Prozessmodell zeigt daher eine relativ kurze Sequenz mit einem Rücksprung von der letzten zur ersten Aktivität. Die Metriken sollten diese Eigenschaft durch einen geringen Grad an Ereignisdichte erkennen können.

Die ersten beiden Prozesse sind vornehmlich erzeugt worden, um die Merkmale *lokaler* Komplexität in deren Ereignisprotokollen aufzudecken. Die folgenden Prozesse beziehen sich hingegen auf die Merkmale *globaler* Komplexität. Um zum Beispiel ein Ereignisprotokoll mit einer besonders großen Trace-Vielfalt zu erhalten, wurde der Prozess zu L_3 erzeugt. Er besitzt eine hohe Anzahl an verschiedenen Prozessvarianten mit jeweils vollkommen unterschiedlichen Ereignissen. Neben einer hohen Trace-Diversität sollten die Metriken ebenfalls erkennen, dass der Prozess zwar relativ kurz ist, schließlich gibt es keine Zyklen und die Varianten enthalten nur maximal 7 Ereignisse, dass diese einzelnen Varianten aber wegen der vielen unterschiedlichen Ereignisse eine hohe Ereignisdichte besitzen.

Im Gegensatz dazu soll mithilfe des vierten Prozesses gezeigt werden, ob die Metriken auch eine Kombination aus einer hohen Trace-Vielfalt und einer geringen Ereignisdichte aufdecken können. Hierzu wurde in den Prozess neben einigen Alternativen auch ein Zyklus eingebaut. Das resultierende Ereignisprotokoll L_4 müsste also eine höhere mittlere Trace-Länge (MTL) besitzen.

Zuletzt sollte mit dem fünften Prozessmodell die Rand-Disparität untersucht werden. Im Gegensatz zu den anderen Prozessen wurde dieser zweimal mit unterschiedlichen Parametern simuliert, sodass zwei verschiedene Ereignisprotokolle entstanden: Die erste Simulation ergab das Protokoll L_5 , das eine gleichmäßige Verteilung der Start- und Endereignisse besitzt. Mit der zweiten Simulation wurde L_6 erzeugt, das im Gegensatz dazu eine willkürlich verschobene Verteilung besitzt. Die Metriken – genauer: die Rand-Disparität und die gewichtete Rand-Disparität – sollten in der Lage sein, diese Merkmale aufzudecken.

Mithilfe der oben beschriebenen, künstlichen Prozesse wurden die Ereignisprotokolle L_1 - L_6 erzeugt. Diese wiederum wurden mithilfe der Metriken hinsichtlich ihrer Komplexitätsmerkmale untersucht. Im weiteren Verlauf werden die Ergebnisse aufgeführt und diskutiert, die sich nach der Anwendung der Metriken auf die erzeugten Ereignisprotokolle zeigten. Die Tabelle 12 gibt einen Überblick über die Auswertung.

Tabelle 12: Auswertung der Komplexitätsmetriken für die künstlichen Ereignisprotokolle L_1 - L_6

ID	MTL	NMTL	ED (%)	RD	GRD	TV (%)	TD (%)
L_1	13	13	100	0	0	7	16
L_2	13	7	54	0	0	22	189
L_3	6	6	100	0	0	84	91
L_4	11	8	73	0	0	60	149
L_5	4	4	100	33	30	67	48
L_6	4	4	100	33	15	67	48

Die Metriken zu Log L_1 zeigen erwartungskonform eine hohe lokale Komplexität. Genommen weist das Log eine relative weite Länge sowie eine maximale Ereignisdichte, weil sich jedes Ereignis in den einzelnen Traces auf eine andere Aktivität bezieht. Ebenfalls erwartungskonform zeichnet die globale Komplexität ein umgekehrtes Bild: Die Rand-Disparität ist minimal, weil es nur genau ein Start- und ein Endereignis gibt. Die Trace-Vielfalt ist ebenfalls minimal und aufgrund der nur sehr wenigen alternativen Pfade ist auch die Trace-Disparität sehr niedrig. Zusammengefasst erkennen die Metriken, dass der Prozess zwar einigermaßen lang ist und eine hohe Ereignisdichte hat, aber kaum komplizierte Varianten enthält. Solche Werte deuten auf einen strukturierten Prozess hin, der – ggf. nach einer stärkeren Fokussierung auf die wesentlichen Aktivitäten – mit traditionellen Process Mining-Werkzeugen unterstützt werden kann.

Die Auswertung der Metriken zum Ereignisprotokoll L_2 zeigt, dass der Prozess zwar ebenfalls relativ lang ist, seine Ereignisdichte aber mit nur 54% relativ gering ist. Das ist nicht verwunderlich, denn die hohe Trace-Länge von 13 resultiert aus dem eingebauten Zyklus, nicht etwa aus Sequenzen mit jeweils neuen Aktivitäten. Auch die minimale Rand-Disparität weist nicht auf einen Case-Management-Prozess hin. Allerdings erzeugt der Zyklus aber auch – wenn auch nur zu einem geringen Anteil – Unterschiede zwischen den einzelnen Traces, die je nach Messmethode unterschiedlich hoch ausfallen (TV=22 und TD=189). Dass die Trace-Disparität so hoch ist liegt daran, dass die Manipulationskosten in der Levenshtein-Distanz zwischen einem einfachen und einen durch den Zyklus verdoppelten Trace hoch sind. Genauer gesagt sind im Vergleich zur mittleren Trace-Länge im Schnitt fast doppelt so viele Manipulationen nötig, um einen beliebigen Trace in einen anderen beliebigen Trace zu verwandeln.

Das Protokoll L_3 sollte gemäß den Erwartungen relativ kurze Traces enthalten, die dafür aber sehr unterschiedlich sind. Diese Annahmen wurden von den Metriken bestätigt: Einerseits ist die mittlere Trace-Länge mit einem Wert von 6 recht kurz. Andererseits sind sowohl die Trace-Vielfalt als auch die Trace-Disparität jeweils sehr hoch (84% und 91%). Die Metriken können also erkennen, ob ein Prozess viele unterschiedliche Varianten besitzt. Darüber hinaus weisen die Rand-Disparitäten an diesem Beispiel auch darauf hin, dass diese Varianten nicht bereits zu Beginn beziehungsweise bis zum Ende auftreten, sondern dass sie erst im Verlauf des Prozesses entstehen. Falls diese Vielfalt an einem zu breiten Scope bei der Protokollierung liegt (zum Beispiel, weil prozessfremde Aktivitäten nicht ausgeschlossen wurden), oder ob die Vielfalt tatsächlich darin begründet ist, dass jeder Fall unterschiedlich ist, muss im Einzelfall geklärt werden. Hierbei hilft eine weitere Beobachtung der Fälle: Sollten keine neue Alternativen

hinzukommen, handelt es sich vermutlich um einen zu breiten Scope. Kommen stets neue Alternativen hinzu, handelt es sich vermutlich um einen komplexen Prozess, der von der adCM-Methode profitieren könnte.

Die Ergebnisse aus den Experimenten mit Protokoll L_4 sollten zeigen, dass die Metriken auch bei einem Mix aus unterschiedlichen Prozesseigenschaften funktionieren. Tatsächlich erkennen die Metriken, dass der Prozess im Verhältnis zu seiner durchschnittlichen Länge von 11 Ereignissen nur 8 unterschiedliche Ereignisse enthält, also eine eher mäßige Ereignisdichte von 73% aufweist. Bezogen auf die globale Komplexität ist das Ergebnis ebenfalls mäßig: Einerseits ist die Trace-Vielfalt eher durchschnittlich (60%), andererseits ist die Trace-Disparität verglichen mit $\log L_3$ sehr hoch (149%). Das liegt daran, dass durch den eingebauten Zyklus sehr hohe Kosten für die Manipulation der Zeichenketten in der Levenshtein-Distanz entstehen. Weil der Zyklus nur in einem von drei alternativen Pfaden auftritt, ist die Trace-Disparität geringer als im zweiten $\log L_2$, bei dem der Zyklus in einem von nur zwei Pfaden vorkommt.

Ob ein Ereignisprotokoll im Vergleich zu seiner Länge viele oder wenige Start- beziehungsweise Endereignisse besitzt, soll die Rand-Disparität messen. Am Beispiel der Ereignisprotokolle L_5 und L_6 konnten die Metriken diese Fähigkeit nachweisen: Bei L_5 , dem Protokoll mit gleichverteilten Start-/Endereignissen ist die Rand-Disparität und deren gewichtete Variante vergleichsweise hoch (RD=33%, GRD=30%). Dass sie nicht weit über 50% geht liegt daran, dass sie relativ zur Länge gemessen wird. Auf diese Weise drückt sie aus, dass die Start- und Endereignisse einzeln betrachtet im Schnitt etwa ein Drittel der Ereignisse im Log ausmachen. Selbstverständlich erzeugen die unterschiedlichen Start- und Endereignisse alternative Pfade durch den Prozess, die durch die anderen beiden Metriken der globalen Komplexität erkannt werden (TV=67%). Die Trace-Disparität ist konform zu den Erwartungen nur mäßig hoch (TD=48%). Bei L_6 werden dieselben Werte gewonnen mit dem Unterschied, dass die gewichtete Rand-Disparität nur etwa halb so groß ist. Dieser Unterschied lässt sich auf die statistische Konzentration auf einige wenige Start- beziehungsweise Endereignisse in L_6 zurückführen.

5.6.2 ERGEBNISSE MIT LOG-DATEIEN AUS DER PRAXIS

Es ist offensichtlich, dass die Experimente mit künstlich erzeugten Ereignisprotokollen nicht ausreichen, um die Validität der Metriken auch auf den Einsatz in der Praxis zu übertragen. Deshalb wurden zusätzlich Experimente mit drei frei verfügbaren Ereignisprotokollen aus der Praxis durchgeführt (siehe Tabelle 13).

Tabelle 13: Details zu den Beispielprotokollen aus den beiden niederländischen Krankenhäusern

ID	Beschreibung	Traces	Ereignisse	Ereignisklassen
L ₇	lang und komplex	1143	150 291	624
L ₈	kurz, aber mit viel Rauschen	2259	24 297	134
L ₉	explizit strukturiert	13087	164 506	23

Das Protokoll L₇ enthält über 1000 Fälle aus einer gynäkologischen Abteilung einer niederländischen Universitätsklinik und wird zusammen mit L₉ von der *IEEE task force on process mining* bereitgestellt (http://data.3tu.nl/repository/collection:event_logs). Es wurde im Rahmen einer Process Mining-Fallstudie angelegt (Mans et al. 2009) und wird seitdem immer wieder in anderen Studien verwendet. Das Protokoll war beispielsweise der primäre Untersuchungsgegenstand bei der ersten Business Process Intelligence Challenge (BPIC), einem jährlich auf der Business Process Management-Konferenz stattfindenden Process Mining-Wettbewerb. Aufgrund seiner Komplexität ist es den Forschern nicht gelungen, ein menschenlesbares Prozessmodell aus den Originaldaten zu extrahieren (siehe Abbildung 33). Stattdessen wurden andere Aspekte wie die Aufschlüsselung beteiligter Organisationen oder ähnlich aggregierte Metainformationen zu den protokollierten Aktivitäten erhoben.

**Abbildung 33: Prozessmodell zu L₇ (modelliert als Transitionssystem)**

L₈ stammt ebenfalls aus einem niederländischen Krankenhaus. Es enthält etwa doppelt so viele Fälle wie L₇, die aus einem Prozess der Intensivstation aufgezeichnet wurden (siehe Abbildung 34). Erste Analysen wurden im Rahmen einer Master-Thesis an der Technischen Universität Eindhoven durchgeführt (Gupta 2007). Es dient als Beispiel-Protokoll³ für einen Discovery-Algorithmus, der speziell auf die Analyse komplizierter Prozesse ausgelegt ist (Weijters und de Medeiros 2006). Darüber hinaus ist das Protokoll aber noch nicht weiter erforscht worden, sodass weniger vergleichbare Mining-Ergebnisse verfügbar sind.

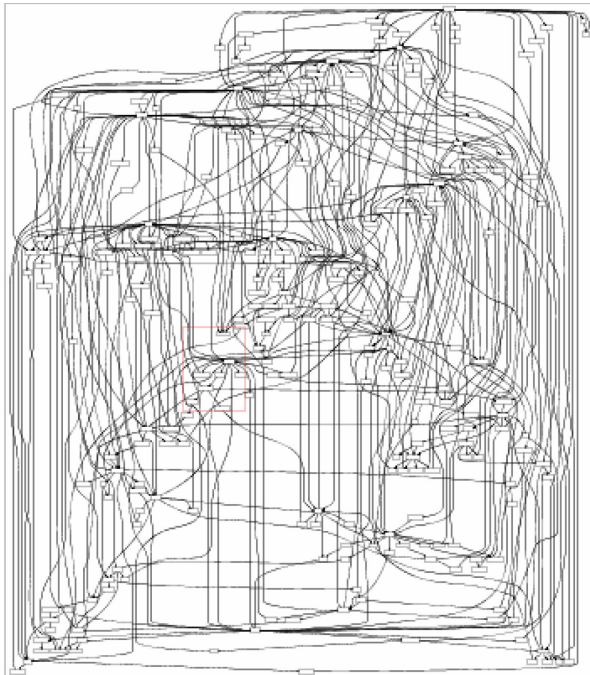


Abbildung 34: Prozessmodell zu L₈ (mit Heuristics Miner vereinfacht)

Das Protokoll L₉ wurde von einem niederländischen Kreditinstitut bereitgestellt und beschreibt Fälle aus einem Prozess zur Prüfung der Kreditwürdigkeit (siehe Abbildung 35). Ebenso wie L₇ war es Gegenstand eines Process Mining-Wettbewerbs, weshalb bereits umfassende Analysen hierzu vorliegen (Rosa und Soffer 2013, S. 217–226).

³ Vgl. https://svn.win.tue.nl/repos/prom/Documentation/UsersGuideHeuristicsMiner6.0/VB_spaghetti_%20illustration3GP.xml

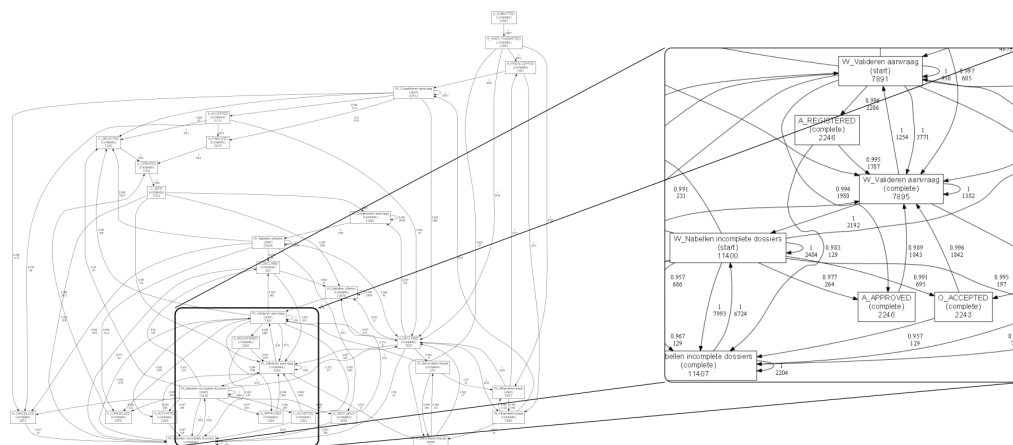


Abbildung 35: Prozessmodell zu L₉ (mit Heuristics Miner vereinfacht)

Erwartungsgemäß müssten die Metriken zeigen, dass die ersten beiden Protokolle im Vergleich zum dritten deutlich komplexer sind. Die errechneten Metriken werden in der Tabelle 14 dargestellt und im weiteren Verlauf detailliert diskutiert.

Tabelle 14: Auswertung der Komplexitätsmetriken für die künstlichen Ereignisprotokolle L₇, L₈ und L₉

ID	MTL	NMTL	ED (%)	RD	GRD	TV (%)	TD (%)
L ₇	131	33	25	1 651	238	95	402
L ₈	15	14	93	60 045	9 062	95	67
L ₉	13	8	62	4	2	65	13

Im Vergleich zu den künstlich erzeugten Prozessen ist auf den ersten Blick zu erkennen, dass das Protokoll L₇ um ein Vielfaches länger ist. Ebenso kann man anhand der Ereignisdichte gut ablesen, dass die Länge maßgeblich durch wiederkehrende Aktivitäten beeinflusst wird. Insgesamt ist die lokale Komplexität von L₇ aber immer noch als relativ hoch einzustufen, weil es im Schnitt immerhin 33 unterschiedliche Aktivitäten pro Fall zu erledigen gilt. Die Metriken zur globalen Komplexität zeichnen ein eindeutiges Bild: Sowohl die ungewichtete als auch die gewichtete Rand-Disparität weisen auf eine sehr hohe Komplexität an den Start- und Endpunkten des Prozesses hin. Außerdem liegt die Wahrscheinlichkeit, dass ein bestimmtes Ereignis in einem beliebigen Trace auftritt, bei nur 5%, sodass die Trace-Vielfalt mit 95% sehr hoch ist. Ein so hoher Wert ist zu erwarten, weil es mit einem Verhältnis von 624 zu 33 viel mehr Ereignisklassen insgesamt gibt als im Mittel pro Trace beobachtet werden. Die anhand der Levenshtein-Distanz gemessene „Unähnlichkeit“ der Traces ist verglichen mit den bisher analysierten Protokollen überragend. Das hängt maßgeblich damit zusammen, dass die Traces sehr unterschiedlich lang sind. Die Spanne reicht von nur einem Ereignis bis über 1800 Ereignisse in einem Trace. Es verwundert also nicht, dass die jeweiligen Unterschiede gut ein Vierfaches der mittleren Trace-Länge ausmachen.

Das zweite Protokoll L_8 ist bezogen auf die lokale Komplexität offensichtlich deutlich weniger komplex als L_7 : Die Traces sind im Mittel nicht länger als 15 Ereignisse, wenngleich sie mit einer Ereignisdichte von 93% kaum wiederkehrendes Verhalten aufweisen. Der Vergleich zwischen den beiden Logs hinsichtlich der globalen Komplexität ist hingegen interessanter. Zwar lassen sich Unterschiede in der Trace-Vielfalt nur im Nachkommabereich feststellen, aber bei der Rand- und der Trace-Disparität sind die Ergebnisse quasi gegensätzlich. Obwohl L_8 nur weniger als die Hälfte an Ereignisklassen enthält, gibt es doppelt so viele unterschiedliche Start- und Endereignisse. Unter der Berücksichtigung, dass es sich bei der Rand-Disparität nicht um eine lineare, sondern exponentielle Metrik handelt, lässt sich dieser Zusammenhang in den Metriken ablesen. Die geringe Trace-Disparität ist darauf zurückzuführen, dass die Traces ziemlich kurz sind sie sich diesbezüglich nicht so stark unterscheiden wie es bei Log L_7 der Fall ist. Folglich werden nicht so hohe Kosten bei der Levenshtein-Distanz hervorgerufen.

Das dritte Echtdaten-Protokoll wurde von den Autoren als „explizit strukturiert“ bezeichnet und müsste demnach auch in den Metriken als deutlich weniger komplex identifiziert werden. Ein Blick auf die lokale Komplexität kann diese Erwartung bestätigen: Die Fälle sind zwar nur etwas kürzer als die Fälle aus L_8 (wenngleich L_8 für ein als schwach strukturiert gekennzeichnetes Protokoll schon ausgesprochen kurz ist), aber ihre Ereignisdichte ist mit 62% deutlich geringer. Das heißt, im Durchschnitt wird fast jedes zweite Ereignis in dem Prozess wiederholt. Auch die globale Komplexität hebt sich deutlich von den anderen Protokollen ab. Insbesondere die Rand-Disparität ist sehr gering. Das lässt sich darauf zurückführen, dass es in dem Prozess nur ein einziges Start-Ereignis und nur zehn mögliche End-Ereignisse gibt. Auch die Trace-Vielfalt ist deutlich geringer. Dass sie nicht noch kleiner ausfällt ist damit begründbar, dass der Prozess im Verhältnis zur mittleren Anzahl unterschiedlicher Ereignisklassen je Trace (NMTL = 8) relativ viele Ereignisklassen insgesamt enthält (23). Die Trace-Disparität weist ebenfalls erwartungskonform auf eine geringe Komplexität hin.

5.7 ZUSAMMENFASSUNG

Dieses Kapitel hat sich mit dem Monitoring befasst, dem ersten Lösungsbaustein der Dissertation. Die Kernaufgabe dieses Bausteins ist das Protokollieren von Benutzerinteraktionen in schwach strukturierten Prozessen, die durch das adCM-Konzept unterstützt werden sollen. Aus dieser Aufgabe heraus entstehen einige Herausforderungen und Anforderungen. Beispielsweise müssen die erhobenen Daten insofern vollständig sein, als dass alle nachfolgenden Lösungsbausteine darauf aufsetzen können. Ebenso müssen die Daten gemäß dem Stand der Forschung eine gewisse

Qualität aufweisen, das heißt sie müssen zum Beispiel wohlstrukturiert und systematisch ermittelt sowie semantisch verknüpft werden.

Im zweiten Unterkapitel wurde untersucht, ob und wie der aktuelle Forschungsstand diese Herausforderungen berücksichtigt und inwiefern sich diese Arbeiten von den basalen Konzepten des adCM unterscheiden. In diesem Zusammenhang wurde herausgestellt, dass sich die adCM-Ansätze vor allem in dem Punkt abgrenzen, dass die Protokollierung einheitlich, zentral und auf einem passenden Abstraktionsniveau erfolgt. Die Ereignisse müssen also nicht mehr aufwändig konsolidiert, abstrahiert oder gefiltert werden und reflektieren gezielt die Kernaktivitäten des Case Managers.

Im Hauptteil des Kapitels wurden die Anforderungen mithilfe dedizierter Lösungskonzepte adressiert und in das übergeordnete adCM-Konzept integriert. Die wesentlichen Beiträge sind...

- die standardkonforme, anwendungsübergreifend einheitliche und systematische Protokollierung der Kernaktivitäten mithilfe eines universellen Datenmodells.
- die semantische Verknüpfung von Ereignisdaten mit den Konzepten des adCM-Metamodells sowie beliebiger weiterer Konzepte mithilfe eines graphbasierten Speicherkonzepts.
- die Identifikation des Fallkontexts mithilfe von Kontexttermvektoren, sodass in späteren Schritten ähnliche Fälle mit potentiell nützlichen Artefakten oder Agenda-Einträgen vorgeschlagen werden können.
- die Beschreibung grundlegender Architekturentscheidungen, mithilfe derer die oben genannten Bestandteile integriert werden können.
- die Verknüpfung der adCM-Methodik mit dem Process Mining mithilfe der Komplexitätsmetriken. Sie sollen den Unterstützungsbedarf durch adCM-Konzepte messbar machen und dadurch als eine Art „Weiche“ zwischen adCM und Process Mining dienen.

Die anschließende Evaluation der Bausteine anhand einer Fallstudie und einer prototypischen adCM-Implementierung hat gezeigt, dass bis auf wenige Ausnahmen die Benutzerinteraktionen vollständig, verlässlich, wohldefiniert und mit domänenspezifischen Konzepten verknüpft protokolliert werden. Ferner konnte gezeigt werden, dass die Strategie zur Ermittlung des spezifischen Fallkontextes dazu herangezogen werden kann, die Ähnlichkeit von Fällen zu messen und zu bewerten. Zuletzt wurde positiv evaluiert, ob die Komplexitätsmetriken nützlich sind, um bestimmte Charakteristiken schwach strukturierter Prozesse erkennen zu können und auf diese Weise dazu beitragen können, sich zu einem gegebenen Ereignisprotokoll für eine geeignete Form

der IT-Unterstützung (Process Mining vs. adCM) zu entscheiden. Für weitere Details zur Implementierung dieser Lösungsbausteine wird auf Kapitel 7 verwiesen.

6 DISCOVERY

Basierend auf den im Monitoring protokollierten Aktivitäten des Case Managers ist es Gegenstand des Discovery-Bausteins, Wissen über den Geschäftsprozess zu sammeln und in einem Modell zusammenzufassen, sodass es dem Case Manager über eine kontextsensitive Unterstützungsfunktion bereitgestellt werden kann (siehe Abbildung 36). Gemäß dem adCM-Ansatz werden keine kontroll- oder datenflussorientierten Modelle, sondern Templates eingesetzt. Die Algorithmen zur Erzeugung von Templates und die Datenstrukturen zur Speicherung ebendieser sind Gegenstand dieses Kapitels.

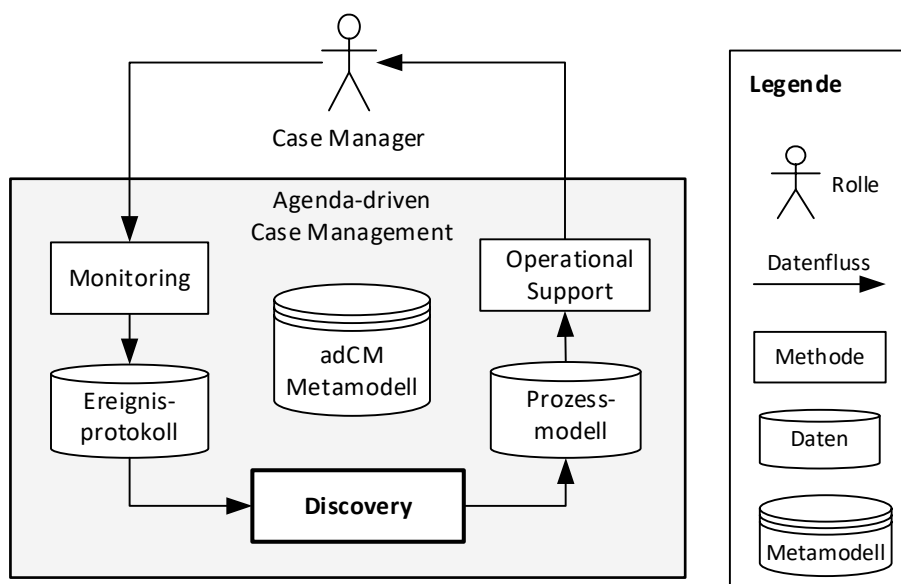


Abbildung 36: Lösungsbaustein „Discovery“ im Gesamtkontext

Analog zum Aufbau von Kapitel 5 werden zunächst die Anforderungen an das Modell sowie Anforderungen an die Discovery-Algorithmen festgelegt, die dieses Modell erzeugen sollen. Passend dazu werden anschließend verwandte Lösungsansätze aus der Literatur diskutiert. Für die noch offen gebliebenen Fragen werden dann eigene Lösungskonzepte zur Anreicherung des Modells vorgestellt.

6.1 PROBLEMSTELLUNG

Dieses Unterkapitel beginnt mit Anforderungen zur Erzeugung von Templates, gefolgt von einer Auflistung von Anforderungen an die Datenbasis zur Speicherung von Templates.

6.1.1 ANFORDERUNGEN AN DAS AUFFINDEN VON TEMPLATES

In diesem Abschnitt wird beschrieben, welche Anforderungen an das Konzept gestellt werden, um Templates zu definieren.

Anforderung 13: Templates manuell definieren

Aufgrund seiner Erfahrungen kann ein Case Manager in der Lage sein, häufig wiederkehrende Aktivitäten beziehungsweise benötigte Artefakte zu identifizieren und in einem Template zusammenzufügen mit dem Ziel der Wiederverwendung durch ihn selbst, durch einen Kollegen oder durch eine Funktion zur kontextsensitiven automatischen Bereitstellung von Templates. Bezogen auf das Fallbeispiel könnte Herr Müller beispielsweise die für einen Oberschenkelhalsbruch typischen und in Form von Agendaeinträgen festgehaltenen Aktivitäten auswählen und zusammen mit einer selbst erstellten Liste renommierter Fachärzte für die entsprechende Therapie als Template ablegen. Falls er selbst oder einer seiner Kollegen die Rehabilitation eines Patienten mit einer vergleichbaren Diagnose managen werden, können sie auf das Template zurückgreifen. **Das adCM-Konzept soll deshalb eine Funktion definieren, die die manuelle Definition dieser Templates ermöglicht.**

Anforderung 14: Templates automatisch erzeugen

Um Case Manager in der Definition von Templates zu unterstützen, soll das adCM-Lösungskonzept eine Funktion zur automatischen Erzeugung von Templates enthalten. Um die Qualität der Templates sicherzustellen (zum Beispiel hinsichtlich ihrer praktischen Relevanz), sollen sie auf Grundlage der protokollierten Benutzerinteraktionen aus vergangenen Fällen erzeugt werden. Treten bestimmte Agendaeinträge oder Artefakte immer in einer bestimmten Konstellation auf, soll dieses Muster erkannt und zusammen mit den manuell erstellten Templates in einer Datenbank abgelegt werden. Wird in der Reha-Management-Abteilung von Herrn Müller beispielsweise nach einer stationären Behandlung sehr häufig auch eine ambulante Physiotherapie durchgeführt, so soll dieses Muster erkannt und in einem Template beschrieben werden.

6.1.2 ANFORDERUNGEN AN DIE SPEICHERUNG VON PROZESSWISSEN

Anforderung 15: Entwicklung eines anpassbaren Datenmodells

Aufgrund der Unplanbarkeit des Kontroll- oder Datenflusses in schwach strukturierten Prozessen führt der adCM-Ansatz Konzepte ein, die ein hohes Maß an Flexibilität bieten sollen. Die Agenda beispielsweise lässt einem Case Manager alle Freiheiten bei der Strukturierung und Ausgestaltung der Aktivitäten. Zudem können alle erdenklichen Artefakte zugeordnet werden (Webseiten, Dokumente, Notizen etc.). Daraus ergeben sich Anforderungen an die Anpassbarkeit der Datenstrukturen zur Ablage von Instanzen dieser adCM-Konzepte. Am Beispiel der unterschiedlichen Artefakttypen bedeutet das unter anderem, dass das adCM-Metamodell um domänenspezifische Typen erweiterbar sein soll. Damit das adCM-Werkzeug beispielsweise einen Heilverlaufsplan von Herrn Müller als solchen identifizieren kann, muss dieser Artefakttyp vorher definiert und an die domänenübergreifende Typhierarchie des adCM-Metamodells angebunden worden sein (zum Beispiel durch den Prozessverantwortlichen). Weil die Artefakttypen aber nicht a priori vollständig definiert und klassifiziert werden können, muss eine solche Ergänzung der Typhierarchie auch im laufenden Betrieb möglich sein. **Deshalb soll ein anpassbares Datenmodell entwickelt werden, das die nachträgliche Erweiterung um domänenspezifische Modellelemente unterstützt.**

Anforderung 16: Synonyme identifizieren, um später bei der operativen Unterstützung ähnliche Fälle besser erkennen zu können

Gemäß des adCM-Konzepts ist es eine bewusste Entscheidung, den Case Managern keine Einschränkungen bei der Benennung von Agendaeinträgen und Artefakten oder beim Verschlagworten von Artefakten vorzuschreiben, wenngleich solche Einschränkungen durch Geschäftsregeln bei Bedarf hinzugefügt werden können. Diese Entscheidung hat allerdings den Nachteil, dass die Bezeichnungen einem starken "Begriffsrauschen" (Noise) unterliegen. Dieses Rauschen verringert die Qualität der Discovery-Algorithmen: Ohne Weiteres kann das System nicht wissen, dass "Heilplan" nur eine andere Schreibweise eines bestimmten Case Managers ist, der eigentlich einen "Heilverlaufsplan" meint. Weil beide Begriffe also getrennt voneinander gezählt werden, unterschätzt der Discovery-Algorithmus deren Relevanz. **Deshalb soll der Discovery-Algorithmus in der Lage sein, Synonyme zu identifizieren und bei der Erzeugung von Templates zu berücksichtigen.**

Anforderung 17: Die einheitliche Verwendung von Domänenbegriffen fördern

Synonyme in der Datenbasis zu identifizieren und durch etablierte Begriffe aus der Domäne zu ersetzen, ist ein wichtiger Anwendungsfall. Nichtsdestotrotz kann es in bestimmten Situationen wichtig sein, das Entstehen dieser Synonyme im Speziellen beziehungsweise dieses Rauschens im Allgemeinen von vornherein zu verhindern oder zumindest zu reduzieren. Eine entsprechende Funktion besitzt außerdem den Vorteil, dass das automatische Ersetzen der Begriffe durch meistverwendete Synonyme zu Irritationen oder gar Missverständnissen beim Einsatz der gemeinsamen Templates führen kann. Herr Müller würde beispielsweise zwar mit hoher Wahrscheinlichkeit verstehen, dass Heilplan und Heilverlaufsplan dieselbe Bedeutung haben: dass HVP aber dasselbe meint, könnte unter den Case Managern umstritten sein. Im schlimmsten Fall würde ein Case Manager ein vorgeschlagenes Template nur deshalb nicht annehmen, weil er – entgegen der Mehrheit seiner Kollegen – bestimmte Begriffe darin nicht in diesem Kontext kennt oder verwendet. **Um solche Differenzen zu vermeiden, soll das adCM-Lösungskonzept eine Funktionalität zur Förderung der einheitlichen Verwendung von Domänenbegriffen enthalten.**

6.2 RELATED WORK

Interessante Beiträge zum Thema „Process Discovery“ stammen aus der Process Mining-Community. Es handelt sich dabei im Wesentlichen um Mining-Algorithmen, die auf die Analyse strukturierter Geschäftsprozesse ausgerichtet sind und kontrollflussorientierte Prozessmodelle erzeugen. Sie werden im weiteren Verlauf dieses Abschnittes vorgestellt.

6.2.1 DISCOVERY-ALGORITHMEN IM PROCESS MINING

Bezugnehmend auf die in Kapitel 3.2 aufgeführten drei Typen des Process Mining ist adCM insbesondere mit dem Discovery-Schritt (dt.: Erkennung) vergleichbar. Nach (van der Aalst 2011) besteht diese Phase aus der Anwendung von Algorithmen, die aus einem zentralen, bereits vorgefilterten Ereignisprotokoll ein Prozessmodell generieren. Der älteste Discovery-Algorithmus ist der bereits im Kapitel 2 kurz erwähnte α -Algorithmus (van der Aalst et al. 2003). Er ist aufgrund seiner kurzen Formulierung leicht zu verstehen und unterstützt zentrale Konzepte wie Nebenläufigkeit. Im Gegenzug liefert er keine guten Ergebnisse bei rauschbehafteten oder unvollständigen Ereignisdaten. Aufgrund dieser Schwächen eignet sich der α -Algorithmus und einige seiner Erweiterungen nur für gut strukturierte Prozesse, die in vollständigen und rauschfreien Ereignisprotokollen abgebildet sind (Rubin et al. 2007; Li et al. 2007; A. K. A. de Medeiros et al. 2004).

Um die Erzeugung von Prozessmodellen auch unter schwierigeren Bedingungen zu ermöglichen, haben Process Mining-Forscher in den letzten zehn Jahren viele weitere Algorithmen entwickelt. Im Gegensatz zum α -Algorithmus sind manche auch explizit für die Anwendung auf weniger strukturierte Prozesse entworfen worden. Um die Ergebnisse der unterschiedlichen Algorithmen vergleichbar machen zu können, wurden einheitliche Qualitätskriterien erforderlich. Besonders weit verbreitet sind die in (Buijs et al. 2012) veröffentlichten Kriterien Fitness, Simplicity, Generalization und Precision. Sie werden daher auch im Evaluationsteil dieses Lösungsbausteins eingesetzt, um die adCM-Lösungsansätze im Detail mit ausgewählten Process Mining-Algorithmen zu vergleichen (siehe Kapitel 6.4).

6.2.2 *DISCOVERY-ALGORITHMEN VARIABLER GESCHÄFTSPROZESSE*

Neben den klassischen Discovery-Ansätzen gibt es darüber hinaus Ansätze wie in (Weidlich et al. 2011), die ein gemeinsames Prozessmodell aus verschiedenen Prozessvarianten erzeugen können. Hierzu werden Fingerabdrücke von Prozessen definiert, um dann mithilfe mengentheoretischer Verfahren Varianten in ähnlichen Geschäftsprozessen aufzuzeigen. Dieser Ansatz ist dem Grundgedanken des adCM sehr nahe, häufig wiederkehrende Muster in Geschäftsprozessen zu identifizieren und in Form von Templates zu beschreiben. Dennoch setzen diese Arbeiten voraus, dass sich die untersuchten Prozesse vollständig als kontrollflussorientiertes Prozessmodell abbilden lassen. Demnach eignen sie sich nicht für schwach strukturierte Prozesse.

6.2.3 *TREE MINING: DISCOVERY-ALGORITHMEN FÜR HIERARCHISCHE DATENSTRUKTUREN*

Die adCM-Konzepte Agenda und Template sind hierarchische Datenstrukturen. Eine Analyse solcher Datenstrukturen auf gemeinsame Elemente ist demnach vergleichbar mit der Suche nach häufig wiederkehrenden Teilbäumen in einem „Wald“ aus Bäumen. Solche Ansätze finden sich unter dem Begriff „**Frequent Subtree Mining**“ (FSM) und dienen ursprünglich der Analyse von XML-Dateien. Aufgrund der weiten Verbreitung von XML gibt es viele Anwendungsfälle für solche Analysen. Entsprechend findet sich in der Literatur auch eine Vielzahl unterschiedlicher Algorithmen. Um die Auswahl eines auf das adCM-Lösungskonzept passenden Algorithmus nachvollziehen zu können, werden in diesem Abschnitt die theoretischen Grundlagen dieser Algorithmen erläutert.

FSM ist eine Methode zur Gewinnung von häufig auftretenden Teilbäumen in einer übergeordneten Baumstruktur. Es handelt sich dabei um eine spezielle Variante zur Lösung des mathematisch übergeordneten Problems „**Frequent Subgraph Mining**“ (Jiang et al. 2013). Viele

FSM-Algorithmen sind auf einen speziellen Anwendungsfall zugeschnitten. Das heißt, je nachdem, wie genau die zu verarbeitenden Bäume strukturiert sind und nach welchen Regeln die resultierenden Teilbäume erzeugt werden sollen, gibt es spezielle Varianten von FSM-Algorithmen. Um einen für den konkreten Anwendungsfall passenden Algorithmus auszuwählen, müssen die folgenden Kriterien geklärt werden (Jiang et al. 2013):

1. **Struktur der zu verarbeitenden Eingangsdaten:** Hier ist unter anderem zu unterscheiden, ob es sich entweder um Wurzelbäume oder freie Bäume handelt und ob die Knoten der Bäume entweder geordnet sind oder nicht.
2. **Struktur des Ergebnisses:** Hier wird unterschieden, ob die Muster induzierte oder eingebettete Teilbäume sind und ob sie maximal sind.

Struktur der zu verarbeitenden Eingangsdaten: In der Graphentheorie unterscheidet man zwischen Bäumen mit und ohne Wurzelknoten. Man spricht in diesem Fall von „freien“ und in jenem von „gewurzelten“ Bäumen oder „Wurzelbäumen“ (Krumke und Noltemeier 2012). Außerdem wird unterschieden, ob die Bäume geordnet sind oder nicht. Wenn die Ordnung eine Rolle spielt, hat das die konkrete Auswirkung, dass zwei Teilbäume mit einer unterschiedlichen Reihenfolge der untergeordneten Knoten nicht als identisch gewertet werden und demnach bei der Mustererkennung nicht zu einem gemeinsamen Muster gezählt werden:

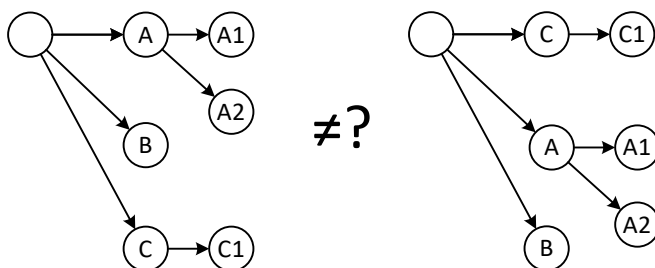


Abbildung 37: Ob diese beiden Bäume gleich sind oder nicht, hängt davon ab, ob sie geordnet sind

Struktur des Ergebnisses: Zuletzt wird ein Algorithmus danach ausgewählt, ob die Templates *induzierte* oder *eingebettete* Teilbäume sein sollen. Induzierte Teilbäume werden nach anderen Regeln erzeugt als eingebettete Teilbäume. Um zu verdeutlichen, wie sich diese Unterscheidung auf das Ergebnis auswirkt, müssen zunächst die charakteristischen Merkmale dieser beiden Teilbaumtypen abgegrenzt werden. Die Abbildung 38 zeigt einen Beispielbaum (a) mit unterschiedlichen Arten von Teilbäumen (b-e).

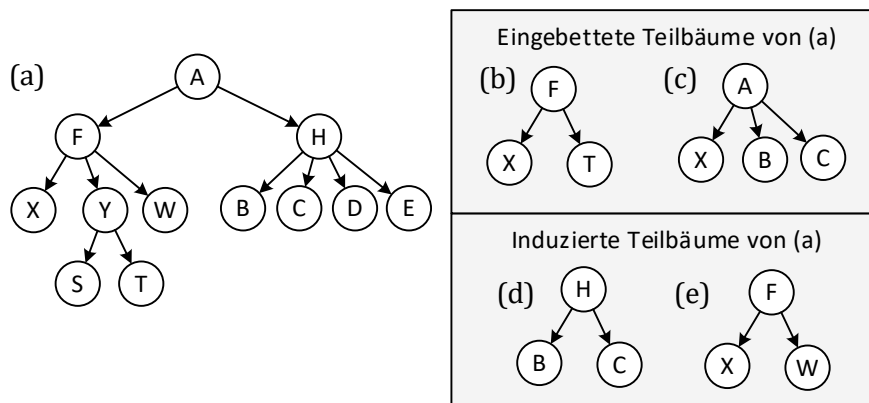


Abbildung 38: Beispielbaum mit Teilbäumen in Anlehnung an (Jiang et al. 2013)

Bei (b) und (c) handelt es sich um eingebettete Teilbäume. Sie werden unter der Annahme erzeugt, dass es sich bei den Abhängigkeiten um transitive Beziehungen handelt. Die derart gewonnenen Teilbäume können demnach bestimmte Knoten in den Pfaden überspringen, sofern sie nicht häufig genug auftreten. Diese Flexibilität führt dazu, dass mehr Teilbäume gefunden werden können als bei der Suche nach induzierten Teilbäumen. Andererseits wird bei diesem Vorgehen aber auch eine möglicherweise in der Hierarchie immanente Semantik, wie zum Beispiel unterschiedliche Abstraktionsebenen, ignoriert.

Die Teilbäume (d) und (e) sind zwei Beispiele für induzierte Teilbäume. Sie berücksichtigen bei der Mustererkennung nur unmittelbare Vorgänger-Nachfolger-Beziehungen, also ohne Transitivität. Diese Art von Teilbäumen ignoriert nicht – im Gegensatz zu den eingebetteten Teilbäumen – eine in der Hierarchie enthaltene Semantik.

Unabhängig davon unterscheiden sich Algorithmen auch dahingehend, ob die erzeugten Teilbäume maximal sind oder nicht. Damit ist gemeint, ob eine minimale Häufigkeit angegeben werden kann, die ein Teilbaum erzielen muss, um zur Ergebnismenge (also dem Template-Wald) zu gehören. Sofern ein Algorithmus diesen Parameter nicht anbietet, berechnet er konsequenterweise die Häufigkeit aller möglichen Kombinationen von Teilbäumen und gibt sie ungefiltert als Ergebnis zurück. Das hat zwei Nachteile: Erstens ist diese Arbeitsweise bei großen Datenmengen nicht performant – schließlich werden selbst solche Teilbäume weiter in der Verarbeitung berücksichtigt, die nur ein einziges Mal auftreten. Zweitens sind diese Algorithmen nicht in der Lage, Anforderungen an die Relevanz der erhobenen Teilbäume zu adressieren. Solche Anforderungen sind beispielsweise dann wichtig, wenn die resultierenden Teilbäume für Recommender Systeme eingesetzt werden.

Eine Analyse der Literatur (Kalinowski 2013) hat folgende Liste an möglichen Algorithmen ergeben:

Tabelle 15: Übersicht über Frequent Tree Mining-Algorithmen

Algorithmus	Eingangsdaten	Ergebnisdaten
AMIOT (Hido und Kawano 2005)	Geordnete Bäume	Induziert
CFFTree (Zhao und Yu 2007)	Freie Bäume	Induziert
Chopper (Wang et al. 2004)	Geordnete Bäume	Eingebettet
CMTreMiner (Chi et al. 2004)	Beides möglich	Induziert, maximal
CousinPair (Shasha et al. 2004)	Ungeordnet	Eingebettet
F3TM (Zhao und Yu 2006)	Freie Bäume	Induziert
FreeTreeMiner (Chi et al. 2003)	Freie Bäume	Induziert
FreqT (Zaki 2004)	Geordnete Bäume	Induziert
FTMiner (Rückert und Kramer 2004)	Freie Bäume	Induziert
HybridTreeMiner (Yun Chi et al. 2004)	Beides möglich	Induziert
IMB3-Miner (Tan et al. 2006b)	Geordnete Bäume	Beides möglich
PathJoin (Chehreghani 2011)	Ungeordnet	Induziert, maximal
PrefixTreeISpan (Zou et al. 2006)	Geordnete Bäume	Induziert
Razor (Tan et al. 2006a)	Geordnete Bäume	Eingebettet
RootedTreeMiner (Chi et al. 2005)	Ungeordnet	Induziert
SLEUTH (Zaki 2004)	Ungeordnet	Eingebettet
TIDS (Tatikonda et al. 2006)	Geordnete Bäume	Induziert
TreeFinder (Termier et al. 2002)	Ungeordnet	Eingebettet, maximal
TreeMiner (Zaki 2002)	Geordnete Bäume	Eingebettet
TRIPS (Tatikonda et al. 2006)	Geordnete Bäume	Induziert
U3 (Hadzic et al. 2010)	Ungeordnet	Induziert
uFreqt (Nijssen und Kok 2003)	Ungeordnet	Induziert
UITree (Chehreghani 2011)	Ungeordnet	Induziert
UNI3 (Hadzic et al. 2010)	Ungeordnet	Induziert
Unot (Jiménez et al. 2008)	Ungeordnet	Induziert
XSpanner (Wang et al. 2004)	Geordnete Bäume	Eingebettet

Welche dieser Algorithmen für das Mining von Templates geeignet sind, wird im Lösungskonzept in Kapitel 6.3.4.2 diskutiert.

6.3 LÖSUNGSKONZEPT

Die im Kapitel „Monitoring“ aufgeführten adCM-Lösungsbausteine befassen sich mit dem Problem, die Benutzerinteraktionen eines Case Managers derart aufzuzeichnen, dass Discovery-Algorithmen Prozesswissen in Form von wiederkehrenden Elementen der Agenda und des Workspace identifizieren und als Templates ablegen können. In diesem Sinne sind die Lösungsbausteine stark voneinander abhängig. Die Möglichkeiten des Monitorings beeinflussen den Erfolg des Discovery und Anforderungen an das Discovery wirken sich teilweise auch auf die Anforderungen an das Monitoring aus. Zum Beispiel sorgt das Datenmodell für die Ereignistypen

dafür, dass das Ereignisprotokoll bereits Ereignisse auf einer fachlichen Ebene enthält. So ist es nicht mehr nötig, technische Ereignisse auf der Ebene von Mausklicks o.ä. mit Methoden des Complex Event Processing zu höherwertigen Ereignissen zu aggregieren. Die nachfolgenden Discovery-Algorithmen können daher ohne zusätzliche Vorverarbeitungsschritte auf den Ereignisinformationen aufsetzen. Umgekehrt wurde während der prototypischen Erprobung der Discovery-Algorithmen unter anderem erkannt, dass die Ereignisdaten aus Performanzgründen physisch aufgeteilt werden mussten, um die zu verarbeitende Datenmenge zu reduzieren.

6.3.1 SEMANTISCHES DISCOVERY

Aufgrund der vielen unterschiedlichen Fälle benötigen Case Manager einen hohen Grad an Flexibilität bei der Ausgestaltung ihrer Aufgaben und beim Sammeln von Informationen. Diese Flexibilität muss sich gemäß Anforderung **15** auch in dem Datenmodell zur Speicherung des Prozesswissens wiederfinden. Um dieser Herausforderung konzeptionell zu begegnen, beschäftigen sich manche Forscher damit, Prozessmodellnotationen mit flexibleren Elementen auszustatten. Ein Beispiel hierfür ist das eingangs beschriebene Case Handling (Aalst et al. 2005). Das adCM-Konzept geht allerdings von der Annahme aus, dass auch diese zusätzlich hinzufundierte Flexibilität nicht ausreicht. Diese Annahme begründet sich in der Argumentation, dass der grundlegende Gedanke eines nicht-deklarativen Modells immer dazu führen wird, den Case Manager in seiner Kreativität einzuschränken. Deshalb verfolgen das adCM-Lösungskonzept im Allgemeinen und die darin definierten Discovery-Algorithmen im Speziellen einen in diesem Kontext bewährten Paradigmenwechsel (Benner-Wickner et al. 2015b). Er lehnt die bislang in Informationssystemen übliche „Closed-World Assumption“ ab, welche bezogen auf den zu unterstützenden Geschäftsprozess besagt, dass nur Aktivitäten erlaubt sind, die vorab in einem Modell definiert wurden (normativ). Stattdessen verfolgt das adCM-Konzept eine „Open-World Assumption“, welche besagt, dass alles erlaubt ist, was nicht explizit in einem Modell verboten wurde (deklarativ). Gemäß diesem Paradigma sind kontrollflussorientierte Prozessmodelle eine weniger geeignete Datenstruktur zur Speicherung von Prozesswissen. Um den deklarativen Ansatz zu unterstützen, eignen sich vielmehr Regelwerke, die die fachlichen Grenzen des ansonsten offenen Lösungsraums für die Arbeitsweise der Case Manager stecken.

Weil der oben skizzierte Paradigmenwechsel eng verwandt ist mit einigen grundlegenden Prinzipien des Semantic Web, soll in diesem Abschnitt genauer beleuchtet werden, inwiefern Semantic Web und adCM sich konzeptionell überschneiden und welche Technologien dabei in adCM adaptiert werden können, um Prozesswissen zu erheben und zu speichern. Hierzu wurden zwei Konzepte in die engere Auswahl genommen: Linked Data (Berners-Lee 2006) und die Open-World-Assumption (Reiter 1978).

6.3.1.1 Linked Data

Menschen begreifen neue Konzepte, indem sie sie mit bereits erlernten Begriffen und deren Eigenschaften in Beziehung setzen (Schermer 2014, S. 169–199). Stellt sich durch einen Abgleich der Eigenschaften in einer Kommunikationssituation heraus, dass es eine große Übereinstimmung zwischen den Assoziationen der Kommunikationsteilnehmer gibt, sprechen sie mit hoher Wahrscheinlichkeit von demselben Begriff.

Damit Maschinen diese Fähigkeit nachbilden können, benötigen sie eine vergleichbare Funktion zur vernetzten Speicherung von Begriffen. Im Forschungsfeld des Semantic Web werden URIs (Uniform Resource Identifier) zur eindeutigen Identifizierung von Begriffen eingesetzt. URIs können je nach Präfix durch verschiedene Unterarten repräsentiert werden, wie zum Beispiel URLs oder E-Mail-Adressen (`http:` beziehungsweise `mailto:`) (Berners-Lee et al. 2005). Im Semantic Web werden sie durch sogenannte „properties“ (dt. Eigenschaften) miteinander in Beziehung gesetzt. Die daraus resultierenden Begriffsnetze sind Graphen, die mit der Sprache RDF (Resource Description Framework) beschrieben und ausgetauscht werden (W3C Recommendation). Das Listing 4 zeigt ein einfaches Begriffsnetz am Beispiel eines Heilverlaufsplans. Die URIs sind durch einen Namensraum-Präfix abgekürzt. Gemäß der RDF-Syntax sind die Informationen in Form von Tupeln organisiert, die aus einem Subjekt (links), einem Prädikat (Mitte) und einem Objekt (rechts) bestehen. Demnach bilden Artefakte und Dokumente jeweils eine Begriffsklasse, wobei Dokumente einen Spezialfall eines Artefakts darstellen. Diese Sachverhalte spiegeln im Übrigen einen Teilausschnitt aus dem adCM-Metamodell wider. Die letzten beiden Zeilen besagen, dass ein Heilverlaufsplan ein spezieller Typ von Dokument ist. Bei diesen Informationen handelt es sich um domänenspezifische Ergänzungen des adCM-Metamodells.

```
PREFIX adcm: <http://se.paluno.uni-due.de/adcm>
PREFIX rehab: <http://accident.insurance.com/rehab>

adcm:Artifact rdf:type :Class .
adcm:Document rdf:type :Class .
adcm:Document rdfs:subClassOf adcm:Artifact .
rehab:HealingPlan rdf:type :Class .
rehab:HealingPlan rdfs:subClassOf adcm:Document .
```

Listing 4: Einfaches Begriffsnetz zu einem Heilverlaufsplan in RDF

Auf diese Weise können Maschinen ihre zunächst jeweils eigenverantwortlich gepflegten Informationen miteinander vergleichen oder austauschen. Über diese eher auf die Persistenz und die Kommunikation von Informationen bezogenen Aspekte hinaus bieten Technologien des

Semantic Web auch Analysemöglichkeiten, die sich für eine Adaption in die adCM-Lösungsbausteine anbieten. Zum Beispiel können unter dem Begriff „Reasoner“ zusammengefasste Softwaresysteme die bestehenden Informationen analysieren mit dem Ziel, daraus abgeleitetes Wissen hinzuzufügen oder die Informationen auf Konsistenz zu prüfen. Dabei wird in der Regel auf die Mächtigkeit von Prädikatenlogik erster Stufe zurückgegriffen. Am oben skizzierten Beispiel wäre demnach ein Reasoner in der Lage, die Transitivität in dem Begriffsnetz zu „verstehen“ und daraus zu schließen, dass ein Heilverlaufsplan ein Artefakt ist. Solche Möglichkeiten sind für das adCM-Konzept deshalb interessant, weil Domänenexperten im praktischen Einsatz von adCM das adCM-Metamodell mit Domänenwissen ergänzen können und die im Monitoring aufgezeichneten Daten auf einer fachlich präzisen Ebene analysiert werden können. Das Konzept von Linked Data und die darauf aufbauenden Technologien ermöglichen, dass solche Anpassungen jederzeit und im laufenden Betrieb der Software möglich sind, ohne – wie im Umfeld relationaler Datenbanken üblich – komplizierte Manipulationen an einem starren Datenschema riskieren zu müssen (Curino et al.). Ein solches Maß an Flexibilität ist vielversprechend für die gezielte IT-Unterstützung von Case Management-Prozessen. Linked Data spielt deshalb eine wesentliche Rolle bei der Speicherung und Verarbeitung von Prozesswissen in adCM.

In der Problemstellung des Monitoring-Lösungsbausteins wurde bereits darauf eingegangen, dass die für das Discovery benötigten Ereignisse mit einer Semantik versehen werden müssen. Das adCM-Lösungskonzept hat diese Anforderung im Kapitel 5.3.2 so adressiert, dass die Ereignisse mit einem Objektmodell verknüpft werden, das stets den aktuellen Zustand des eines Falls beziehungsweise der darin eingesetzten Instanzen des adCM-Metamodells repräsentiert. Weshalb dieses Objektmodell aber ausgerechnet graphbasiert gespeichert wird, ist in dem Abschnitt noch nicht motiviert worden. Die hier beschriebenen Vorteile der Semantic Web-Technologien bilden die Grundlage für diese Entwurfsentscheidung.

6.3.1.2 Open World Assumption

Bei der Abfrage von Daten aus klassischen Informationssystemen gehen Benutzer davon aus, dass die abgefragte Datenbasis ein konsistentes Abbild der realen Geschäftsobjekte enthält. Liegt beispielsweise zu einem Versicherungskunden kein Lebensversicherungsvertrag im Bestand vor, geht ein Versicherungsmakler davon aus, dass ein solcher Vertrag (noch) nicht existiert. Eine solche „closed world assumption“ ist in gut erforschten Domänen etabliert. Die Ursache dafür ist, dass zu Beginn eines solchen Prozesses entweder alle Informationen bereits bekannt sind – vergleichbar mit Ebene 0 der 5 Ebenen der Unwissenheit (Armour 2000) – oder es zumindest

keine besondere Herausforderung darstellt, alle relevanten Informationen für einen Prozess zusammenzutragen (Ebene 1: „Ich weiß, was ich nicht weiß“).

Anders verhält es sich in Domänen wie dem Gesundheitssektor oder ganz bestimmten Geschäftsbereichen der Unfallversicherung, bei denen die zu bearbeitenden Sachverhalte oder Probleme so beschaffen sind, dass kaum im Vorfeld abgeschlossen geklärt werden kann, welche Informationen überhaupt zur Bearbeitung benötigt werden (Ebene 2: „Ich weiß nicht, was ich nicht weiß“). In solchen Fällen können Abfragen des Datenbestandes prinzipiell nur unvollständige Ausschnitte einer stetig wachsenden Wissensbasis sein. Dieses Denkmodell zur Interpretation von Daten wird als „open world assumption“ verstanden (Reiter 1978).

Übertragen auf adCM bedeutet dies, dass bei der Analyse der im Monitoring beobachteten und in den Ereignisprotokollen festgehaltenen Aktivitäten des Case Managers andere Regeln gelten als bei klassischen Informationssystemen: Lassen sich beispielsweise bestimmte Aussagen wie „ein Patient mit einem komplizierten Bruch wird nach der stationären Behandlung mit einer Physiotherapie behandelt“ in den Daten nicht finden beziehungsweise nicht ableiten, so bedeutet dies nicht, dass eine solche Therapie falsch oder ungültig ist, sondern dass sie schlicht noch nicht beobachtet wurde. Folglich lässt sich keine Aussage über den Wahrheitsgehalt eines solchen Sachverhaltes treffen. Nach ebendiesen Gesetzen soll also auch der adCM Discovery-Algorithmus arbeiten.

Die beiden oben vorgestellten Semantic Web-Konzepte beziehen sich auf technologischer Ebene sowohl auf die Speicherung des durch die Discovery-Algorithmen erhobenen Prozesswissens als auch auf die Discovery-Algorithmen selbst. Im nächsten Unterkapitel wird zunächst beleuchtet, wie sie in die adCM-Lösungskonzepte zur Speicherung von Prozesswissen einfließen.

6.3.2 MEHRSCHICHTIGES DATENMODELL FÜR DAS DISCOVERY

Um die adCM-Konzepte für eine „intelligente“ Unterstützung des Case Managers einsetzen zu können, werden zwei Dinge benötigt: eine Datenbasis, die Wissen über den Prozess enthält und Algorithmen, die auf dieser Wissensbasis arbeiten. In diesem Unterkapitel wird beschrieben, mit welchen Design-Entscheidungen die Anforderungen an die Wissensbasis beantwortet werden. Dabei wird auch auf die Nützlichkeit diverser Semantic-Web-Technologien eingegangen.

Gemäß der oben motivierten Adaption von Semantic Web-Technologien sieht das adCM-Lösungskonzept zur flexiblen Speicherung von Prozesswissen eine graphbasierte Ablage der wesentlichen Informationen vor. Diese Datenstruktur ist entlang des Geltungsbereiches der darin

gespeicherten Informationen in fünf logisch miteinander verbundenen Schichten aufgeteilt (siehe Abbildung 39).

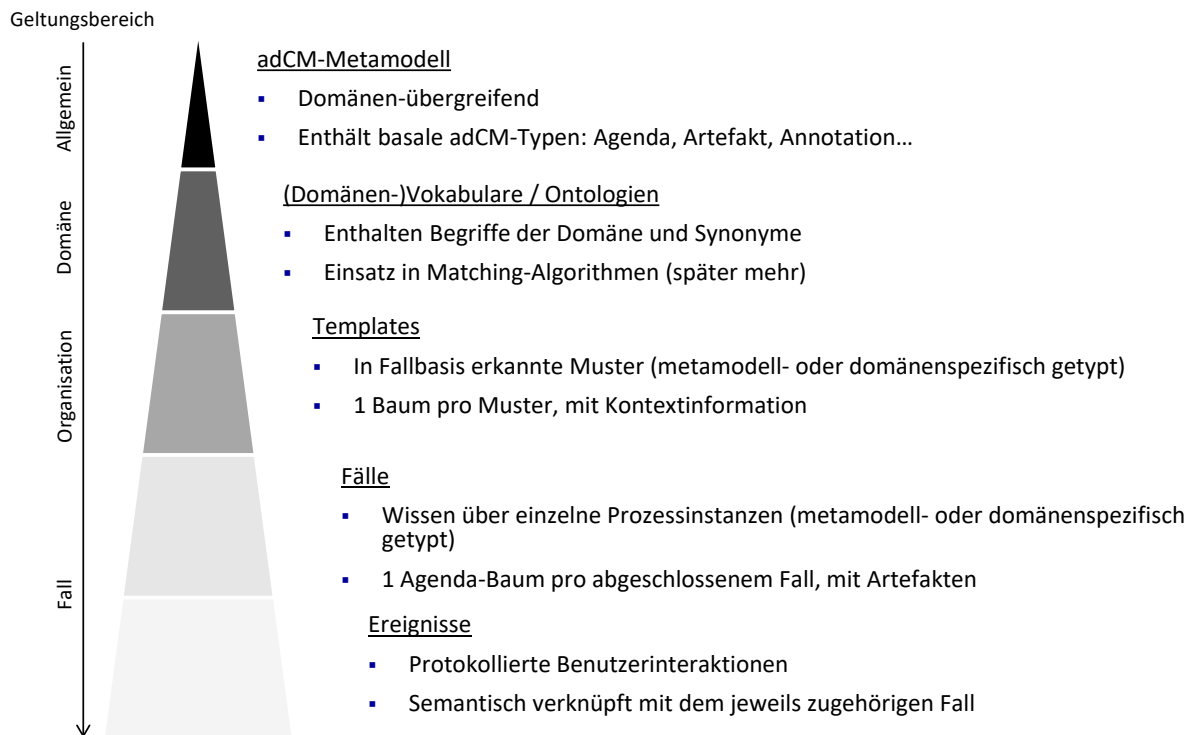


Abbildung 39: Mehrschichtiges Datenmodell für das Discovery

Die oberste Schicht ist das adCM-Metamodell. Streng genommen reflektiert diese Ebene noch kein Prozesswissen. Vielmehr werden darin mithilfe der Beschreibungssprache OWL/DL formal nur die abstrakten domänenübergreifenden Typen (Agenda, Artefakt, Dokument, etc.) definiert (siehe Kapitel 6.3.2.1). Um diese notwendigerweise unvollständige Menge an Typen bezogen auf den zu unterstützenden Prozess zu präzisieren, können Domänenvokabulare beziehungsweise domänenspezifische Ontologien zusätzliche Begriffe und deren Zusammenhänge aus der Domäne bereitstellen. Sie werden üblicherweise ebenfalls unter Zuhilfenahme von Semantic Web-Technologien definiert und bilden die zweite Ebene des adCM-Datenmodells (siehe Kapitel 6.3.2.2). Die Templates sind das Ergebnis der adCM Discovery-Algorithmen. Ihre Elemente sind Instanzen von Klassen des adCM-Metamodells oder einer domänenspezifischen Ontologie. Weil Templates in jedem Unternehmen individuell erstellt werden, sind sie organisationsspezifisch und bilden somit die dritte Ebene (siehe Kapitel 6.3.2.3). Ursprung dieser Templates ist der Bestand an Falldaten. Ebenso wie die Templates handelt es sich bei den Falldaten um Instanzen von adCM- oder domänenspezifischen Klassen. Allerdings sind sie fallspezifisch und bilden demzufolge die vierte Ebene (siehe Kapitel 6.3.2.4). Auf der fünften und letzten Ebene befinden

sich die Ereignisdaten, die wie im Monitoring-Baustein beschrieben mit den Falldaten semantisch verknüpft werden.

Die Bausteine des mehrschichtigen Datenmodells werden in den nachfolgenden Kapiteln einzeln im Detail beschrieben.

6.3.2.1 Formalisierung des AdCM-Metamodells

Das bereits in Kapitel 4 eingeführte adCM-Metamodell definiert die wesentlichen Konzepte des Agenda-driven Case Management. Es enthält ausschließlich Begriffe, die auf die besonderen Merkmale von Case Management-Prozessen zugeschnitten sind. Domänenspezifische Begriffe werden bewusst herausgehalten, damit die damit verknüpften Methoden übergreifend in sehr unterschiedlichen Domänen wie dem Reha-Management oder dem Software Engineering eingesetzt werden können. Somit enthält das Metamodell noch kein Prozesswissen.

Die darunterliegenden Schichten des Datenmodells verwenden die Konzepte aus dem Metamodell, um die Informationen zu typisieren. Das Metamodell übernimmt damit die Aufgabe einer übergeordneten Strukturbeschreibung ähnlich einem Datenbankschema. Um die Vorteile der Semantic Web-Technologien zu nutzen, wurde zur Beschreibung des Modells die Logik-basierte Sprache OWL-DL ausgewählt. Neben den Vorzügen der Flexibilität ist sie insbesondere auch dafür geeignet, Regeln für die Struktur der Instanzen (Fälle, Templates etc.) festzulegen und somit eine Grundlage für die Überprüfung der Fallbasis zu schaffen. Während sich die Regeln auf dieser Ebene nur auf die Verwendung der allgemeinen Typen des adCM-Metamodells beziehen, können in unteren Schichten mit OWL-DL auch domänen- oder organisationsspezifische Regeln formuliert werden. Details hierzu wurden in (Benner-Wickner et al. 2015b) veröffentlicht.

Um das adCM-Metamodell zu formalisieren wurden zuerst RDF-Klassen für alle bestehenden Konzepte wie Agenda, Fall etc. definiert. Jeder Klasse wurden Eigenschaften (Properties) zugewiesen, die wichtige Informationen zu der Klasse beschreiben. Zum Beispiel kann einem Fall ein Boolean zugewiesen werden, der festlegt, ob er abgeschlossen ist. Artefakte hingegen haben ein Erzeugungsdatum usw. Neben der Definition der Klassen selbst und deren potentieller Eigenschaften werden in dem OWL-Metamodell auch die Beziehungen zwischen den Klassen definiert. Ein Agendaeintrag kann beispielsweise beliebig viele Artefakte sowie eine Liste von untergeordneten Agendaeinträgen enthalten. Diese semantischen Beziehungen zwischen Konzepten werden ebenfalls mithilfe von OWL-Properties beschrieben. Das nachfolgende Listing zeigt in der ersten Zeile ein Beispiel, wie eine solche Property in OWL definiert wird.

```
adcm:containsArtifact rdf:type owl:ObjectProperty .  
adcm:containsArtifact rdfs:domain adcm:AgendaItem .  
adcm:containsArtifact rdfs:range adcm:Artifact .
```

Listing 5: Auszug aus dem formalisierten adCM-Metamodell

Ohne Weiteres würde die erste Zeile nicht verhindern, dass diese Eigenschaft jedem beliebigen Subjekt und Objekt zugewiesen werden kann. Es wäre also durchaus erlaubt – wenngleich semantisch falsch – über „containsArtifact“ zu definieren, dass ein Dokument ein Artefakt enthält. Ebenso wäre es aber auch erlaubt, und das ist viel gravierender, einem Agendaeintrag über „containsArtifact“ ein Artefakt zuzuweisen, obwohl das assoziierte Objekt gar kein Artefakt ist, sondern zum Beispiel ein Agendaeintrag. Solche falschen Zuordnungen beeinflussen die durch den Reasoner durchgeführten Schlussfolgerungen und können dadurch die Analyseergebnisse verfälschen. Deshalb werden die beiden nachfolgenden Zeilen hinzugefügt. Sie legen genau fest, welche Klassen als Subjekt und Objekt infrage kommen. Bezogen auf die Open World Assumption werden also bereits hier auf höchster Ebene die Grenzen des Handlungsspielraums eines Case Managers festgelegt. Alles, was nicht deklarativ mithilfe von OWL spezifiziert wurde, ist prinzipiell erlaubt.

6.3.2.2 Domänenspezifische Ergänzung des adCM-Metamodells

Auf der zweiten Ebene befinden sich die Domänen-Vokabulare oder domänenspezifischen Ontologien, die mithilfe des Linked Data-Konzepts das formalisierte adCM-Metamodell ergänzen. Sie bilden die zentrale Voraussetzung zur Adressierung der Anforderungen **15**, **16** und **17**. Denn um zum Beispiel das Begriffsrauschen zu vermeiden, was unweigerlich durch die Freitexteingabe bei der Benennung und Verschlagwortung von Agendaeinträgen eintritt, können Domänenvokabulare eine wichtige Rolle spielen. Im medizinischen Umfeld bieten sich hierfür Ontologien wie zum Beispiel MeSH an. Sie bietet mit über 218.000 Einträgen die Möglichkeit, Synonyme wie „Vitamin C“ und „Ascorbinsäure“ zu erkennen. Aber auch ähnlich umschriebene Diagnosen können mithilfe einer domänenspezifischen Ontologie wie zum Beispiel dem ICD-Code eindeutig zugeordnet werden. Diese Zuordnung ermöglicht den Discovery-Algorithmen Muster zu finden, obwohl bestimmte Begriffe darin sehr unterschiedlich formuliert werden. Ohne eine solche Funktion würden diese Templates dem Discovery verborgen bleiben. Wie diese Zuordnung im Detail funktioniert, wird im späteren Verlauf des Kapitels beschrieben.

Sofern keine domänenspezifischen Vokabulare zur Identifikation von Synonymen verfügbar sind, gibt es zumindest die Möglichkeit, eine Ontologie mit dem Wortschatz der gewählten Sprache zu verwenden. Beispiele hierfür sind die beiden Lexika WordNet (Fellbaum 2012) und der deutschsprachige Ableger GermaNet (Kunze 2005). Sie enthalten jeweils über 80.000

beziehungsweise 100.000 sogenannte Synsets. Dabei ist aber zu beachten, dass insbesondere im Kontext von wissensintensiven Prozessen eine hohe Anzahl an Fachbegriffen eingesetzt wird, die in solchen allgemeinen Ontologien üblicherweise nicht enthalten sind. Deshalb bietet sich die Erzeugung von Ontologien aus Datawarehouse- oder Intranetsystemen an (Navigli und Velardi 2004).

Aus der technischen Perspektive betrachtet gibt es zwei Strategien zur Integration von bestehenden Ontologien in das adCM-Datenmodell. Einerseits kann dem Linked-Data-Ansatz gemäß die Ontologie physikalisch in die Datenbasis übernommen und mithilfe der bereits bekannten Properties mit den bestehenden Konzepten verknüpft werden. Für eine solche Verknüpfung benötigt man allerdings hohe Personalressourcen, weil diese Arbeiten von Experten aus der Domäne durchgeführt werden müssen. Alternativ zu dieser Strategie kann der Inhalt der Ontologie andererseits auch über einen Service angebunden und nachgeschlagen werden (lookup-Service). Das heißt, ein Discovery-Algorithmus kann auf die Ontologie über eine separate Programmierschnittstelle zugreifen, die Synonyme zu einem bestimmten Begriff liefert. Falls die zu integrierenden Ontologien sehr groß sind, empfiehlt sich die zweite Strategie, weil sonst die insgesamt zu verarbeitende Datenmenge für die ohnehin performanzkritischen (Wang und Parsia 2007) Reasoner zu groß wird. Bei der prototypischen Umsetzung des adCM-Lösungskonzepts wurden die Ontologien deshalb als lookup-Service implementiert.

Neben der Möglichkeit, über Linked Data ganze Ontologien zu verknüpfen, sieht das adCM-Lösungskonzept auch die Option vor, *einzelne* domänen- oder organisationsspezifische Sachverhalte im Model zu verankern. Eine einfache Art dies zu tun ist das Ergänzen von Unterklassen bestehender adCM-Konzepte. Der in Listing 4 vorgestellte OWL-Code zeigt bereits eine domänenspezifische Ergänzung am Beispiel des Heilverlaufsplans. Für diesen Begriff ließen sich zusätzlich mithilfe der OWL-Property „owl:sameAs“ wie folgt verschiedene Synonyme deklarieren (siehe Listing 6).

```
reha:Heilverlaufsplan owl:sameAs reha:HVP  
reha:Heilverlaufsplan owl:sameAs reha:HV-Plan
```

Listing 6: Manuell im Domänenvokabular deklarierte Synonyme

Ein weiterer wichtiger Aspekt einer solchen Deklaration ist im Übrigen auch, dass auf diese Weise nicht nur alle für ein Artefakt gültigen Funktionen mit einem Heilverlaufsplan und dessen synonym verwendeten Objekten möglich sind. Vielmehr können zusätzliche Regeln erstellt werden, die sich konkret auf den Heilverlaufsplan oder seine Synonyme beziehen. Eine solche

Regel könnte beispielsweise fordern, dass ein abgeschlossener Fall immer einen Heilverlaufsplan enthalten muss. Mehr Details über die Deklaration von Geschäftsregeln mithilfe domänenspezifischer Erweiterungen in dieser Ebene des Datenmodells finden sich in (Benner-Wickner et al. 2015b).

Wie einfach auch tiefergehende Manipulationen an dem gewählten Datenmodell sind, zeigt das folgende Beispiel. Angenommen, nach der Integration eines adCM-Werkzeugs in die Anwendungslandschaft wird der Bedarf nach einer Wiedervorlagefunktion festgestellt. Demnach sollen alle Agendaeinträge und Artefakte mit einem verbindlichen Abgabetermin versehen werden können, der vom System geprüft und dem Case Manager kurz vor Ablauf gemeldet wird. Neben den obligatorischen Änderungen an der Benutzerschnittstelle und der Geschäftslogik ist die Änderung am Datenmodell einfach (siehe Listing 7).

```
rehab:dueDate rdf:type owl:ObjectProperty .
rehab:dueDate rdfs:domain adcm:AgendaItem .
rehab:dueDate rdfs:domain adcm:Artifact .
rehab:dueDate rdfs:range xsd:dateTime.
```

Listing 7: Domänenspezifische Ergänzung: Agendaeinträge und Artefakte erhalten ein Abgabedatum

6.3.2.3 Graphbasierte Beschreibung der Templates

Unterhalb der domänenspezifischen Ergänzungen des adCM-Metamodells durch Ontologien oder Vokabulare befindet sich die Templates-Ebene. Wie schon in Kapitel 4.3.3 kurz eingeführt, handelt es sich bei den Templates, ähnlich wie der Agenda, um beliebig schachtelbare Baumstrukturen aus Agendaeinträgen und zugeordneten Artefakten. Sie unterscheiden sich von den Agenden nur dahingehend, dass sie fallübergreifendes Prozesswissen repräsentieren und einen gewissen Support haben. Der Support gibt die Anzahl der Fälle an, in denen dieses Template enthalten ist. Das adCM-Lösungskonzept sieht je eine Strategie zur dezentralen und zur zentralisierten Speicherung von Templates vor.

Die dezentrale Speicherung sieht vor, dass die Template-Ebene im übertragenen Sinne ein „Wald“ unabhängiger Template-Bäume ist. Jedes neue Template wird diesem Bestand hinzugefügt, ohne dass andere Templates dadurch beeinflusst werden. Diese Strategie bietet die maximale Flexibilität und integriert sich gut in die oberen Schichten, weil sie in derselben graphbasierten Datenbank abgelegt werden. In Anlehnung an die Anforderung 14 kann diese Strategie aber auch nachteilig sein, sobald sich die Templates untereinander sehr stark überschneiden. In solchen Fällen wäre nämlich die Pflege der Templates ausgesprochen schwierig. Deshalb sieht das adCM-Lösungskonzept alternativ eine stärker normalisierte Speicherung der Templates vor. Das Listing 8 zeigt als Beispiel die Eigenschaften eines Templates, das mit der dezentralen Speicherstrategie

in der graphbasierten Datenbank gespeichert wurde. Es stammt aus den Ereignisdaten eines niederländischen Krankenhauses und wurde erstellt mit dem adCM Miner, der später in diesem Kapitel detailliert beschrieben wird.

SPARQL-Query:

```
PREFIX adcm: <http://paluno.uni-due.de/adcm#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>

describe ?x
where {
  ?x rdf:type adcm:Template
}
LIMIT 1
```

Ergebnis:

```
@prefix adcm: <http://paluno.uni-due.de/adcm#> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .

<http://paluno.uni-due.de/opc/example_templates/AutoGenTemplate205/>
  a adcm:Template ;
  adcm:description "^^xsd:string ;
  adcm:inBusinessProcessType <http://paluno.uni-due.de/opc/example
    /BusinessProcessType/medical_treatment/> ;
  adcm:name "AutoGenTemplate205"^^xsd:string ;
  adcm:rootElement <http://paluno.uni-due.de/opc/example
    /Case/Template/AgendaRootPlaceholder/> ;
  adcm:version 1 .
```

Listing 8: Beispieltemplate des Prozesses „medical_treatment“, gespeichert in der graphbasierten Datenbank

Im Gegensatz zu einem Wald aus unzusammenhängenden, teils redundanten Template-Bäumen, gibt es bei der zentralisierten Speicherstrategie genau ein zentrales großes Template, in das alle Templates aufgehen und das infolgedessen das gesamte Prozesswissen enthält. Welche der darin enthaltenen Knoten für welchen Fallkontext relevant sind, wird mithilfe von Schlüsselworten an die einzelnen Knoten annotiert. Um ein für einen bestimmten Kontext passendes Template aus dem zentralen Template extrahieren zu können, werden die Schlüsselwörter ähnlich einer Facettensuche mit dem Fallkontext abgeglichen und bei Bedarf herausgefiltert. Am Ende der Filterung bleibt von dem zentralen Template nur noch das für den aktuellen Kontext relevante Template zurück. Details zu dieser Strategie werden in (Benner-Wickner et al. 2016) beschrieben.

6.3.2.4 Falldaten

Die nächst-spezifischere Ebene im Datenmodell ist der Bestand an Falldaten. Die darin gespeicherten Informationen beziehen sich jeweils immer auf einen konkreten Fall und werden mithilfe des Meta-Modells und den domänenspezifischen Anpassungen getypt. Ähnlich der dezentralen Variante der Template-Ebene ist die Fall-Ebene ein Wald aus Fall-Bäumen, deren

Wurzelement eine Instanz des Metamodell-Elements „Case“ ist. Jeder Fall hat genau eine Agenda, die wiederum den Wurzelknoten für die Agendabaumstruktur bildet und die über die jeweiligen Workspaces auch alle assoziierten Artefakte enthält.

Das adCM-Lösungskonzept sieht vor, die Fallbasis zwischen abgeschlossenen und offenen Fällen aufzuteilen. Das hat mehrere Vorteile: Erstens dürfen unvollständige Fälle nicht zur Erzeugung von Templates herangezogen werden – schließlich können sie sich aufgrund der gebotenen Flexibilität noch während des Prozessverlaufs stark ändern. Durch die Trennung wird aber nicht nur diese Einschränkung realisiert, sondern zugleich auch die Performanz der Reasoner verbessert, weil sie nicht den gesamten Bestand analysieren müssen. Die Abbildung 21 in Kapitel 5.3.2 zeigte bereits, wie diese Fälle erzeugt werden. Sobald der Benutzer den Fall schließt und das entsprechende Ereignis eintrifft, wird der Fall in die Menge der abgeschlossenen Fälle verschoben.

Die in der Literatur zu findenden Discovery-Algorithmen – hauptsächlich aus dem Process Mining Forschungsgebiet – benötigen neben den Ereignisdaten keine Fallbasis zur Analyse eines Prozesses. Dies liegt darin begründet, dass sie sich auf den sequentiellen Ablauf von Aktivitäten beschränken und die Ereignisdaten hierfür ausreichend sind. Das adCM-Lösungskonzept sieht hingegen vor, die Struktur aus Agenda und Artefakten der abgeschlossenen Fälle zu untersuchen und darin wiederkehrende Muster in Form von Templates zu erheben. Deshalb wird diese Ebene mit den wesentlichen Falldaten zusätzlich zu den Ereignisdaten benötigt. Die Kombination aus beiden Ebenen ermöglicht im Übrigen zusätzliche Erkenntnisse über den untersuchten Prozess. Beispielsweise lässt sich mit unterstützenden Informationen aus den Ereignisdaten untersuchen, ob es unter den Case Managern vielleicht doch einen Konsens über die Abarbeitungsreihenfolge der verwendeten Agendaeinträge eines Templates gibt.

6.3.2.5 Formalisierung der Ereignisdaten

Auf der Ebene der Ereignisse befindet sich die semantisch angereicherte und zentralisiert gespeicherte Version des Ereignisprotokolls. Es wird, wie in Kapitel 5.3.2 beschrieben, von der Case Management-Anwendung erzeugt und enthält alle Benutzerinteraktionen, die sich auf die Konzepte des adCM-Metamodells beziehen. Für Details zur Modellierung des Wissens wird auf das entsprechende Monitoring-Kapitel verwiesen.

6.3.3 MANUELLE ERZEUGUNG VON TEMPLATES

Bevor die adCM Discovery-Algorithmen zur automatischen Erzeugung von Templates beschrieben werden, befasst sich das Kapitel zunächst mit der Definition von Templates per Hand,

zum Beispiel durch den Case Manager (siehe Anforderung **13**). Es handelt sich dabei nicht um eine Alternative, sondern ein Konzept ergänzend zu der automatischen Erzeugung und wurde einleitend auch in (Benner-Wickner et al. 2014b) diskutiert. Die Motivation für eine solche Funktion ist die Annahme, dass der Case Manager ungeachtet aller IT-Unterstützung selbst am besten beurteilen kann, welche Aktivitäten oder Artefakte sich für ein Template eignen.

Es ist zu erwarten, dass das Bedürfnis, ein Template zu erstellen, spontan während der Prozessausführung auftritt, weil beispielsweise das erneute Erstellen einer häufig wiederkehrenden Agendastruktur als störend empfunden wird. Die Spontanität dieser Situation ist vergleichbar mit dem Moment, in dem Smartphone-Benutzer entscheiden, als interessant eingestufte Medien mit anderen Menschen zu teilen. Diese Funktionen sind üblicherweise über nur wenige Schritte erreichbar und gut in den Dialogfluss integriert. Die manuelle Definition von Templates sollte demnach auf ähnlich komfortable Weise in das Bedienkonzept integriert werden.

Das adCM-Lösungskonzept adaptiert zu diesem Zweck das Konzept des „Teilens“ von Templates. Hierzu wird dem Case Manager jederzeit die Möglichkeit gegeben, den aktuellen Zustand der Agendastruktur (oder Ausschnitte davon) als Template abzulegen und mit anderen Case Managern zu „teilen“. Die dafür benötigte Benutzeroberfläche kann zu jedem Zeitpunkt gestartet werden und überlagert die Arbeit am aktuellen Fall nur für den Moment der Templatedefinition. Auf diese Weise wird die Unterbrechung des Arbeitsablaufs auf ein Minimum reduziert.

6.3.4 AUTOMATISCHE DEFINITION VON TEMPLATES

Gemäß Anforderung **14** soll der Discovery-Baustein des adCM-Lösungskonzepts eine Funktion zur automatischen Erzeugung von Templates enthalten. In diesem Sinne handelt es sich um den wichtigsten Baustein dieser Dissertation, denn er liefert einen wesentlichen Beitrag zur Lösung des übergeordneten Problems: der IT-Unterstützung von Case Managern durch das Vorschlagen wiederkehrender Muster während der Prozessausführung.

Das Erzeugen der Templates wird durch einen Algorithmus realisiert, der in (Benner-Wickner et al. 2015a) unter dem Namen adCM Miner eingeführt wurde. Dessen Teilschritte (siehe Abbildung 40) werden im weiteren Verlauf des Kapitels einzeln beschrieben.

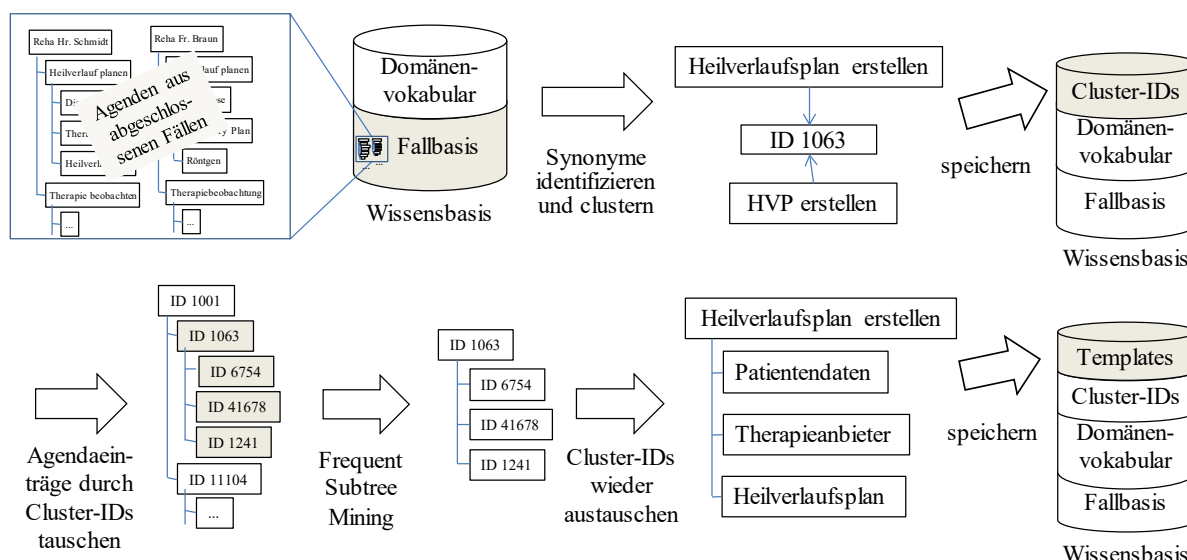


Abbildung 40: Die wesentlichen Verarbeitungsschritte des adCM Miners

6.3.4.1 Vorverarbeitung: Synonyme identifizieren und clustern

Eine große Herausforderung bei der Erzeugung von Templates ist die aus Sicht des Algorithmus ungünstige, aber für das adCM-Konzept zentrale Flexibilität bei der Strukturierung der Agenda. Sie führt zu dem bereits in den vorangegangenen Kapiteln erwähnten Rauschen, weil zum Beispiel die gleiche Aktivität in anderen Fällen unterschiedlich benannt werden kann. Dabei kann es sich um ganz andere Wörter handeln, die synonym zu verstehen sind, aber auch um leichte Varianten in der Schreibweise, hervorgerufen durch unterschiedliche Flexionen oder gar durch Rechtschreibfehler. Die Konsequenz daraus ist, dass der Vergleich der abgeschlossenen Fälle sich in dieser Hinsicht weit komplizierter darstellt als im klassischen Process Mining, bei dem gemäß des im Process Mining-Manifest beschriebenen vierten Grundprinzips darauf vertraut werden kann, dass die Ereignisprotokolle bereits ein einheitliches Vokabular verwenden (van der Aalst et al. 2012).

Der adCM Miner adressiert diese Herausforderung mit zwei Vorverarbeitungsschritten (siehe Abbildung 40): Im ersten Schritt werden alle in der Fallbasis verwendeten Agendaeintragsnamen ermittelt und mithilfe der im Datenmodell verankerten Domänenvokabulare beziehungsweise domänenspezifischen Ontologien auf unterschiedliche Schreibweisen untersucht. Die dadurch entstehenden Wortgruppen werden dabei jeweils einer Cluster-ID zugewiesen und aus Performanzgründen persistiert. Wird der Algorithmus nicht zum ersten Mal ausgeführt, werden die Wörter den bereits bestehenden passenden Gruppen zugeordnet und erhalten somit auch die bereits existierende Cluster-ID. Im zweiten Schritt wird eine für die Dauer des Algorithmus transiente Kopie der abgeschlossenen Agenda-Bäume erzeugt, in der alle Namen durch die zugeordnete Cluster-ID ersetzt werden.

Obwohl die Cluster dem ersten Anschein nach über die Zeit hinweg einigermaßen stabil sein sollten, bietet es sich an, diese Vorverarbeitungsschritte bei jeder Ausführung des Algorithmus zu berücksichtigen. Denn die regelmäßige Aktualisierung der Cluster hilft bei der Lösung eines grundsätzlichen Problems der Prozessanalyse: dem sogenannten „Concept Drift“ (van der Aalst 2011). Dabei handelt es sich um das Phänomen, dass sich Begrifflichkeiten im Laufe der Zeit ändern und das modellierte Prozesswissen dadurch schrittweise ungültig wird, sofern es nicht stets auf dem aktuellen Stand gehalten wird.

Neben den bereits genannten fachlichen Vorteilen hat das Austauschen der Agendaeintragsnamen durch Cluster-IDs auch einen rein technischen Vorteil. Es ist nämlich deutlich performanter, ausschließlich numerische Werte zu verarbeiten und miteinander zu vergleichen als Zeichenketten. Das betrifft erstens den benötigten Speicherplatz, denn die Zeichenketten würden ein Vielfaches an Arbeitsspeicher einnehmen, verglichen mit einer nur vier Byte langen ID, die alleine bereits den gesamten Deutschen Wortschatz referenzieren könnte. Ebenfalls um ein Vielfaches langsamer ist der Vergleich zweier Zeichenketten wegen der erforderlichen Umwandlung in deren entsprechende numerische Repräsentation (gemäß der gewählten Codierung). Trotz dieser Vorteile haben die Cluster-IDs natürlich auch einen ganz offensichtlichen Nachteil: Sie können nicht vom Case Manager verstanden werden und müssen konsequenterweise im letzten Schritt wieder durch ein Wort aus der Wortgruppe ersetzt werden (siehe Kapitel 6.3.4.4).

6.3.4.2 Frequent Subtree Mining: Auswahl des Algorithmus

Bevor auf die Arbeitsweise des FSM-Algorithmus eingegangen wird, muss die Auswahl des Algorithmus motiviert werden. Dies erfolgt auf Basis der in Kapitel 6.2.3 beschriebenen Kriterien:

1. **Struktur der zu verarbeitenden Eingangsdaten**
2. **Struktur des Ergebnisses**

Struktur der zu verarbeitenden Eingangsdaten: Laut adCM-Metamodell besitzt jede Agenda einen Wurzelknoten. Deshalb muss bei der Auswahl des Algorithmus darauf geachtet werden, dass er Wurzelbäume als Eingangsdaten verarbeiten kann. Dann bleibt noch zu klären, ob die Agendaeinträge geordnet sind oder nicht. Zur Erinnerung: Wenn die Ordnung eine Rolle spielt, dann werden zwei Teilbäume mit einer jeweils unterschiedlichen Reihenfolge der untergeordneten Knoten nicht als identisch gewertet und demnach nicht bei der Mustererkennung berücksichtigt. Es gibt zwei Gründe, weshalb Agenden nicht als geordnete Bäume interpretiert werden sollen: Erstens legt das adCM-Konzept nicht fest, in welcher

Reihenfolge die Agenda abgearbeitet werden soll. Zweitens wird dadurch die Anzahl an übereinstimmenden Teilbäumen erhöht, was die Effektivität des Recommenders verbessert. Aus diesen Gründen sind nur diejenigen Algorithmen in der engeren Auswahl, die die Ordnung der Knoten ignorieren.

Struktur des Ergebnisses: Der Algorithmus wird auch danach ausgewählt, ob die Templates *induzierte* oder *eingebettete* Teilbäume der Agenden sein sollen. Zur Erinnerung: Induzierte Teilbäume werden nach anderen Regeln erzeugt als eingebettete Teilbäume (siehe Abbildung 38). Nach eingebetteten Teilbäumen zu suchen bietet mehr Optionen bei der Erzeugung von Templates, weil dabei auch transitive Beziehungen berücksichtigt werden können. Grundsätzlich würde diese Flexibilität zu dem Agenda-Konzept passen. Andererseits ignorieren eingebettete Teilbäume aber eine zentrale Eigenschaft der Agenden, nämlich die in der Hierarchie der Agenda immanente Semantik. An einem Beispiel aus dem Reha-Management kann gezeigt werden, wie das zu einem Problem führen kann: Angenommen es ist unter den Kollegen von Herrn Müller üblich, in der ersten Ebene der Agenda die grundlegenden Therapiemaßnahmen und in der zweiten Ebene deren Zustand zu beschreiben (zum Beispiel offen, begonnen, erledigt, archiviert). Dann würden die Knoten in der ersten Ebene aufgrund der Vielfalt an Maßnahmen insgesamt viel seltener auftreten als die Knoten der zweiten Ebene. Eingebettete Teilbäume erzeugende Algorithmen würden folglich die erste Ebene ignorieren und ein Template ablegen, das die Zustände der Aktivitäten unmittelbar der Agendawurzel zuordnet. Deshalb werden alle Algorithmen ausgeschlossen, die nur eingebettete Teilbäume erzeugen. Stattdessen werden Algorithmen in die Auswahl aufgenommen, die induzierte Teilbäume erzeugen.

Es ist aus Sicht des Case Managers wichtig, dass das Recommender System effizient ist, also nur dann Templates vorschlägt, wenn diese auch relevant sind. Schließlich stellt die Vorschlagsfunktion auch eine – wenn auch kurze – Unterbrechung des Arbeitsflusses dar. Algorithmen auszuwählen, die nur nach maximalen Teilbäumen suchen, kann zu einer höheren Effizienz beitragen. Denn beim Erzeugen maximaler Teilbäume wird (zum Beispiel durch den Prozessverantwortlichen) eine Untergrenze für die Häufigkeit der gefundenen Teilbäume angegeben. Diesen Schwellwert (auch „**Support**“ genannt) müssen die Teilbäume erzielen, um in das Ergebnis aufgenommen zu werden.

Zusammengefasst sind alle Algorithmen aus Tabelle 15 geeignet, die bezogen auf die zu verarbeitenden Daten mit ungeordneten Bäumen arbeiten können und induzierte, maximale Teilbäume finden. Filtert man alle anderen Algorithmen aus der Liste heraus, bleibt die Wahl zwischen dem CMTreeMiner und dem PathJoin-Algorithmus. Eine detailliertere Betrachtung

beider Algorithmen zeigt, dass der PathJoin-Algorithmus eine für den adCM-Kontext ungeeignete Einschränkung besitzt: Er liefert nur dann gültige Ergebnisse, wenn ausgeschlossen werden kann, dass zwei Knoten in einem Baum dieselbe Bezeichnung haben. Unter der Maßgabe höchstmöglicher Flexibilität ist der PathJoin-Algorithmus ebenfalls ausgeschieden, sodass nur der CMTreeMiner als anwendbarer Algorithmus übrigbleibt. Wie im nachfolgenden Abschnitt gezeigt wird, errechnet der Algorithmus nicht die Häufigkeit jedes möglichen Teilbaums, sondern nur der Teilbäume, die das angegebene Mindestmaß an Häufigkeit (Support) erreichen können. Insofern ist er nicht nur der einzig geeignete Algorithmus, sondern auch sehr effizient.

6.3.4.3 Frequent Subtree Mining: Arbeitsweise

Die Idee, die den meisten FSM-Algorithmen zugrunde liegt, ist das "Wachsenlassen" von kleinen, sehr häufig vorkommenden Teilbäumen bis sie so groß sind, dass sie unter den Support-Grenzwert fallen. Abbildung 41 zeigt dieses grundlegende Prinzip an einem kleinen Beispiel-Wald aus drei Bäumen, zu dem alle Teilbäume mit einem Support von mindestens 3 ermittelt werden sollen.

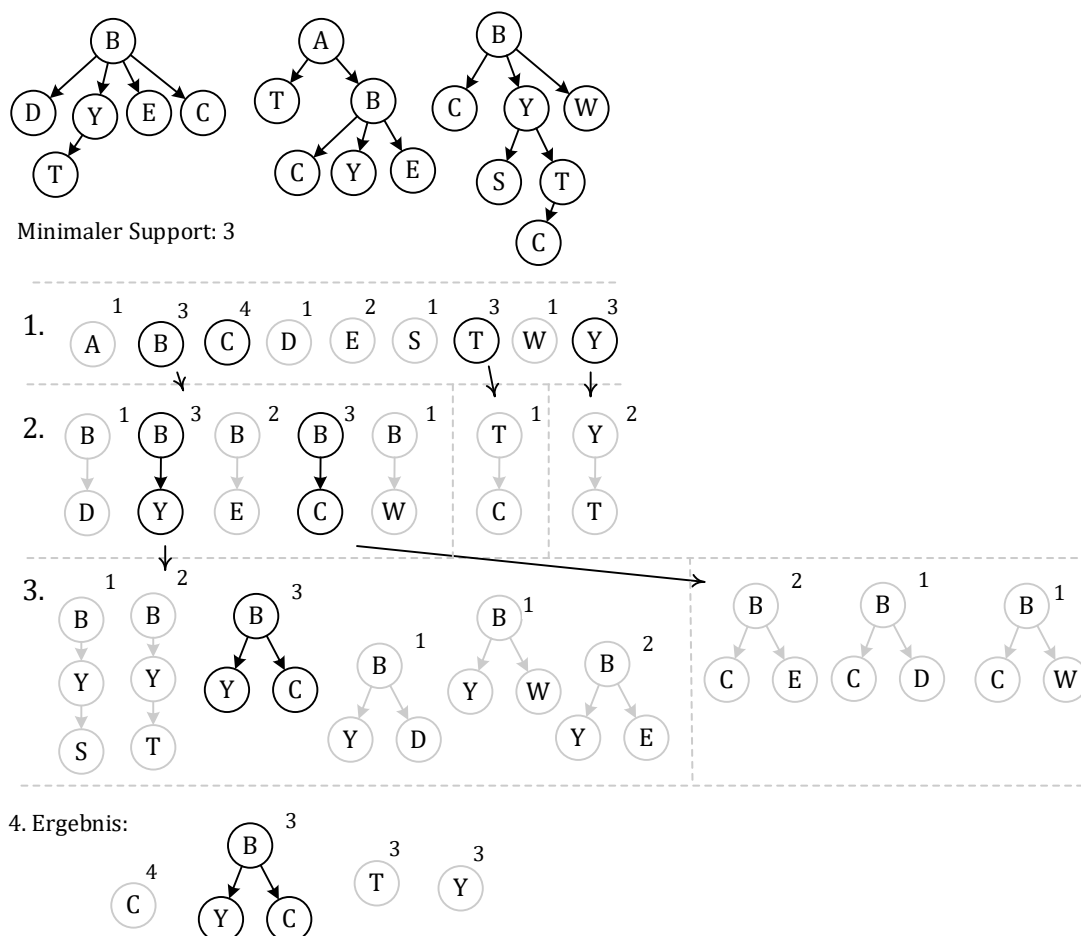


Abbildung 41: Das "Wachsen" von Teilbäumen als grundsätzliches Prinzip von FSM-Algorithmen

Im ersten Schritt werden alle Knoten einzeln hinsichtlich ihres Supports untersucht. Nur diejenigen, die nicht unter die Support-Schwelle (hier: 3) fallen, werden im zweiten Schritt um weitere, im Wald beobachtbare Knoten ergänzt. Der CMTreeMiner beginnt dabei mit den am häufigsten vorkommenden Knoten. Das Hinzufügen von Knoten führt üblicherweise dazu, dass der dadurch entstehende Teilbaum seltener vorkommt. Also muss dessen Häufigkeit neu berechnet werden. Dadurch fallen einige der Teilbäume wieder unter die Support-Grenze und brauchen nicht weiter beachtet werden. Ist ein bestimmter Teilbaum nicht mehr erweiterbar und hat die Schwelle nicht unterschritten, so kann er vorzeitig zum Ergebniswald hinzugefügt werden. Am gezeigten Beispiel trifft das auf die Knoten C, T und Y zu: C ist nicht erweiterbar und die Teilbäume von T und Y fallen unter die Support-Schwelle. Weil die vom CMTreeMiner verwendete Definition eines Teilbaums auch einzelne Knoten einschließt, sind C, T und Y gültige Ergebnis-Teilbäume.

Um den Algorithmus an die Gegebenheiten im adCM-Kontext anzugleichen, muss an dieser Stelle eine Veränderung vorgenommen werden. Denn gemäß dem AdCM-Metamodell sind Templates immer kontextbezogen. Das heißt, sie können nur dann vorgeschlagen werden, wenn der Case Manager sich in einem Kontext befindet, der deren Einsatz rechtfertigt. Ein einzelner Knoten bietet keine Grundlage für eine Kontextinformation. Erst wenn es mindestens einen Elternknoten gibt, kann ein Algorithmus zum Vorschlagen von Templates diese(n) Knoten als Kontext verwenden, mit dem Kontext des Case Managers vergleichen und bei hinreichender Übereinstimmung den Rest des Templates vorschlagen. Als Konsequenz ergänzt der adCM Miner eine minimale Knotenanzahl für die erzeugten Templates.

Der dritte Schritt ist in zweierlei Hinsicht interessant: Erstens wird hier zum ersten Mal deutlich, dass beim Wachsen der Teilbäume zwei verschiedene Strategien angewendet werden können. Sie können entweder zuerst in der Tiefe oder in der Breite erweitert werden. In diesem Punkt unterscheiden sich manche Algorithmen hinsichtlich ihrer Präferenz zu einer der beiden Möglichkeiten. Der CMTreeMiner erweitert den Baum zunächst in seiner Tiefe (depth-first). Zweitens lässt sich im dritten Schritt der Unterschied erkennen zwischen dem CMTreeMiner und Algorithmen, die **nicht** nach maximalen Teilbäumen suchen. Denn prinzipiell fällt der Teilbaum aus dem Wurzelknoten B und dem Kindknoten Y mit einem Support von 3 nicht unter die Schwelle, dennoch zählt er nicht zum Ergebniswald. Das liegt daran, dass dieser Teilbaum nicht maximal ist. Er kann um C erweitert werden, ohne unter die Schwelle zu fallen. Im Kontext von adCM ist das eine vorteilhafte Eigenschaft, weil auf diese Weise der Template-Wald nicht unnötig vergrößert wird. Trotzdem wird dadurch die geschätzte Nützlichkeit der Unterstützungsfunktion (im Sinne der Häufigkeit der angebotenen Templates in der Fallbasis) nicht negativ beeinflusst.

Im Ergebnis ist also das Muster aus B, Y und C erkannt worden. Während die einzelnen Knoten im Ergebniswald des CMTreeMiners enthalten wären, sind sie im adCM Miner ausgeschlossen worden.

Verglichen mit anderen Algorithmen sind der CMTreeMiner im Allgemeinen und der adCM Miner im speziellen nicht nur hinsichtlich der im vorangegangenen Kapitel aufgeführten Klassifikation besser für den Einsatz im adCM geeignet. Darüber hinaus sind die Algorithmen insbesondere bei der Verarbeitung von größeren Teilbäumen deutlich performanter. Experimente zeigten, dass insbesondere ab einer Teilbaumgröße von etwa zehn Knoten die Laufzeit des vergleichbaren PathJoin-Algorithmus im Gegensatz zum CMTreeMiner exponentiell steigt. Es liegt nahe, dass vor allem bei komplizierten Fällen die Zahl der Agendaeinträge diesen Wert schnell überschreitet und der CMTreeMiner deshalb die bessere Wahl ist. Details über die formalen Grundlagen des CMTreeMiners, dessen Arbeitsweise sowie über die Evaluation finden sich in (Chi et al. 2004).

6.3.4.4 Nacharbeiten

Nachdem – wie in Abbildung 40 skizziert – der Algorithmus die Templates unter Angabe des minimalen Supports und der Mindestgröße errechnet hat, müssen die Cluster-IDs wieder zurück in menschenlesbaren Text umgewandelt werden. Das Problem an diesem Schritt ist allerdings, dass die Abbildung der Begriffe auf IDs nicht injektiv ist, schließlich werden bei der Vorverarbeitung mehrere Wörter einem Cluster zugeordnet. Für die Auswahl eines Repräsentanten aus der Menge der möglichen Wörter wird also eine möglichst nachvollziehbare Strategie benötigt. Das adCM-Lösungskonzept folgt dabei einem Konsensprinzip. Das heißt, die Cluster-ID wird durch das Wort ersetzt, welches am häufigsten Verwendung findet. Bei der automatischen Neuanlage von Templates wird diese Strategie bei den Case Managern noch nicht zu Verwirrungen führen. Im schlechtesten Fall werden vorgeschlagene Templates von manchen Case Managern abgelehnt, weil sie die Synonyme anders verstehen. Allerdings sollten die manuell erstellten Templates nicht durch den Algorithmus angepasst werden, zum Beispiel um sie zu optimieren. Denn in einem solchen Fall würde ein Case Manager sein selbst erstelltes Template möglicherweise nicht mehr wiederfinden.

Damit das Konsensprinzip funktioniert, benötigt der adCM Miner Statistiken über die Häufigkeit des Vorkommens aller Wörter in den Agendabäumen. Deshalb werden die Wörter während des Clusterings gezählt und ihre Häufigkeit wird zusätzlich zu der Cluster-ID-Zuordnung persistiert.

6.3.4.5 Zeitpunkt der Ausführung

Weil sich mit jedem abgeschlossenen Fall der Support eines beliebigen Templates ändern kann, sollte der Algorithmus für den Produktivbetrieb so umgestellt werden, dass er inkrementell arbeitet. Das heißt, er muss nicht den gesamten Bestand an abgeschlossenen Fällen verarbeiten, sondern nur jeweils die Veränderungen durch einen jüngst abgeschlossenen Fall analysieren und betroffene Templates umgestalten. Solange der Algorithmus aber nicht inkrementell arbeitet, bieten sich zwei verschiedene Strategien an, nach denen der Discovery-Algorithmus angestoßen werden kann. Eine Alternative ist, diese Berechnungen zu festgelegten, regelmäßigen Zeitpunkten (zum Beispiel nachts oder am Wochenende) als Batch auszuführen. Dieser Ansatz hat den Vorteil, dass die bereitgestellten Rechenressourcen effizienter genutzt werden können und genug Zeit für zusätzliche Vorbereitungsmaßnahmen wie zum Beispiel eine Sicherheitskopie des Datenbestandes bleibt. Sobald ein Fall von einem Case Manager abgeschlossen wird und das entsprechende Ereignis ausgelöst wurde, wird der Discovery-Algorithmus angestoßen.

6.4 EVALUATION⁴

In diesem Kapitel wird anhand einer Fallstudie überprüft, ob mit dem vorgestellten Discovery-Algorithmus Templates gewonnen werden können und mithilfe von etablierten Metriken eingeschätzt, wie nützlich diese Templates sind. Die Resultate werden mit Ergebnissen verglichen, die von traditionellen Discovery-Algorithmen aus dem Process Mining produziert wurden.

6.4.1 EXPERIMENTAUFBAU

Die Evaluation des adCM Miners und der dazugehörigen Komponenten verfolgt zwei Ziele: Erstens soll im Sinne einer Machbarkeitsstudie gezeigt werden, dass die beschriebenen Algorithmen funktionstüchtig sind. In diesem Sinne soll überprüft werden, ob sie überhaupt in der Lage sind, Templates aus einer Menge an abgeschlossenen Fällen zu identifizieren, sodass sie im Falle einer Kontextübereinstimmung während eines laufenden neuen Falls vorgeschlagen werden können. Es wird also für die Evaluation des adCM Miners bereits die Komponente zum Vorschlagen von Templates benötigt. Diese wird allerdings in diesem Kapitel nur als Blackbox betrachtet und später in Kapitel 7 detailliert beschrieben. Als zweites Ziel soll mit der Evaluation aber auch gezeigt werden, dass der gewählte Ansatz nützlich ist. Hierzu sollen die Ergebnisse den Ergebnissen vergleichbarer Ansätze aus der Process Mining-Forschungsdisziplin gegenübergestellt werden.

⁴ Dieses Thema wurde einleitend auch in Benner-Wickner et al. 2015a diskutiert

Um das Erreichen der genannten Ziele feststellen zu können, ist die Durchführung einer Vergleichsstudie geeignet. Eine solche Studie ist jedoch in diesem Zusammenhang nur schwer zu realisieren. Denn hierzu müsste sich eine Organisationseinheit dazu bereit erklären, eine ganze Reihe von Fällen langfristig mit mindestens zwei unterschiedlichen Ansätzen zu unterstützen. Auch eine Interventionsstudie wäre zwar zielführend, setzt aber eine umfangreiche Zeitplanung und Organisation voraus, die in diesem Kontext durch eine einzelne Dissertation schwer leistbar ist. Deshalb ist die Evaluation so entworfen worden, dass eine Art automatisiertes „offline“-Experiment durchgeführt wird. Im Sinne einer Fallstudie basiert es auf der Verarbeitung von Ereignisdaten, die in der Praxis aufgezeichnet wurden. Wie in Abbildung 42 skizziert, wird das entsprechende Ereignisprotokoll im Rahmen des Experiments sowohl von den adCM-Komponenten verarbeitet (oberer Datenfluss) als auch von gezielt ausgewählten, gut erforschten Algorithmen aus dem Process Mining (unterer Datenfluss). Weil es keine ausreichend große Datenquelle mit abgeschlossenen Fällen in Form von Agenden gibt, werden offiziell verfügbare Ereignisprotokolle der IEEE Task Force on Process Mining eingesetzt. Mithilfe einer Adapterkomponente werden diese sequentiellen Ereignisströme nach verschiedenen, im weiteren Verlauf detaillierter beschriebenen Strategien in eine hierarchische Agendastruktur transformiert, sodass der adCM Miner damit arbeiten kann. Weil die Nützlichkeit der Templates bekanntermaßen kontextbezogen ist, müssen sie zuerst in einem Kontext vorgeschlagen werden, bevor die Nützlichkeit anhand von Metriken gemessen wird. Und weil es sich um ein automatisiertes Experiment mit einem Datensatz von mehreren hundert Fällen handelt, muss auch die Auswahl der vorgeschlagenen Templates automatisiert werden. Hierzu wird eine Komponente benötigt, die den Case Manager bei der Template-Auswahl vertritt. Auch hierzu wurden unterschiedliche Strategien entwickelt und erprobt, um die Einflussnahme der simulierten Komponenten zu untersuchen. Um die Zielerreichung zu dokumentieren und mit den Process Mining-Algorithmen abzugleichen, werden im letzten Schritt der beiden Simulationspfade Qualitätsmetriken errechnet. Die Metriken beziehen sich auf Qualitätsaspekte, die in der Process Mining-Literatur üblich sind.

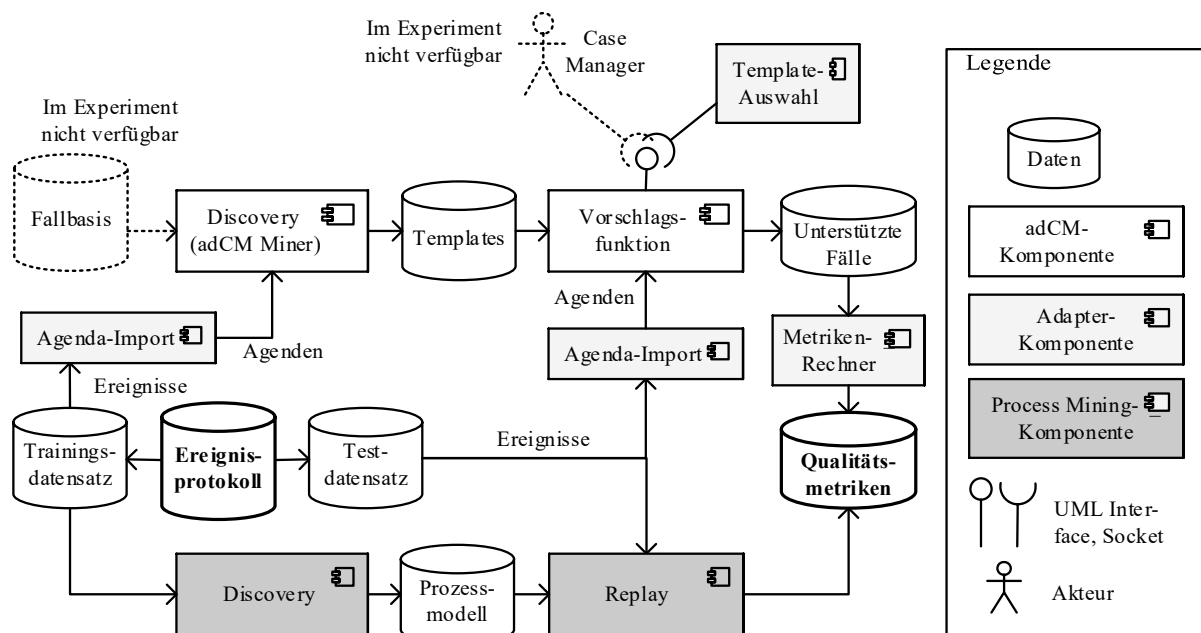


Abbildung 42: Datenfluss des Experiments, beginnend mit dem Ereignisprotokoll, bis zum Vergleich der Qualitätsmetriken (oben: adCM-Teststrecke; unten: Process Mining-Teststrecke)

In den folgenden Abschnitten wird entlang von sieben Schritten erläutert, wie die Experimente vorbereitet und durchgeführt wurden (siehe Abbildung 43).

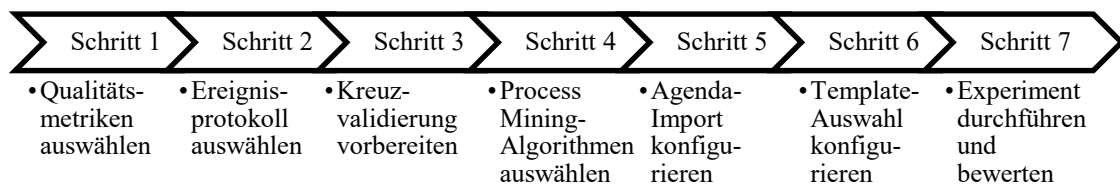


Abbildung 43: Schritte zur Vorbereitung und Durchführung der Evaluation

6.4.1.1 Qualitätsmetriken auswählen

In diesem Schritt werden Metriken ausgewählt, die die Nützlichkeit des adCM Miners und der im Experiment eingesetzten Vorschlagsfunktion messbar machen sollen. Damit die Ergebnisse möglichst gut vergleichbar sind, werden keine gänzlich neuen Metriken eingeführt, sondern bestehende Metriken aus der Process Mining-Disziplin übernommen. In der Literatur werden hierzu vorrangig die vier Metriken Fitness, Precision, Generalization und Simplicity verwendet (van der Aalst 2011; Buijs et al. 2012). Die ersten beiden Metriken stammen aus dem Forschungsgebiet des Information Retrieval und werden zuerst vorgestellt.

Precision (Präzision) beschreibt, wie gut das Modell zum Prozess passt. Im Kontext des Process Mining ist die Präzision ein Maß dafür, inwiefern ein erzeugtes Prozessmodell zu viel Spielraum für Verhalten erlaubt, das eigentlich nicht Teil des Prozesses sein soll. Wenn beispielsweise das Modell viele Elemente enthält, die gar nicht in der Praxis beziehungsweise in einem

aufgezeichneten Ereignisprotokoll auftreten, dann ist das Modell nicht präzise. Falls umgekehrt jedes Element im Modell auch in der Praxis beobachtbar ist, dann ist das Modell maximal präzise. Die Präzision als Qualitätsmetrik hinzuzuziehen ist deshalb wichtig, weil dadurch unerwünschte Phänomene entdeckt werden können, die metaphorisch auch als „Flower Model“ bezeichnet werden (siehe Abbildung 48). Dabei handelt es sich um Modelle oder Modellteile, die zwar in allen anderen Qualitätsmetriken perfekt abschneiden, aber jedes Verhalten rechtfertigen, das in den „Blättern“ abgebildet ist. Untersucht man beispielsweise die Qualität unterschiedlicher Modelle zur Beschreibung des unten abgebildeten Ereignisprotokolls, so hat das Beispielmmodell zwar die positiven Eigenschaften, dass es ...

- a) ... jeden der vier Traces abbilden kann (perfekte Fitness).
- b) ... in Zukunft nur dann angepasst werden muss, wenn neue Ereignisse beziehungsweise Aktivitäten hinzukommen (hohe Generalization).
- c) ... sehr einfach nachzuvollziehen ist (hohe Simplicity).

Umgekehrt hat es aber auch den Nachteil, dass es sehr viele Traces „akzeptiert“, die vielleicht gar nicht zum Prozess gehören. Denn das Modell erklärt alle möglichen Traces, die aus den Ereignissen A bis H bestehen. Demnach ist das abgebildete Modell **unpräzise**.

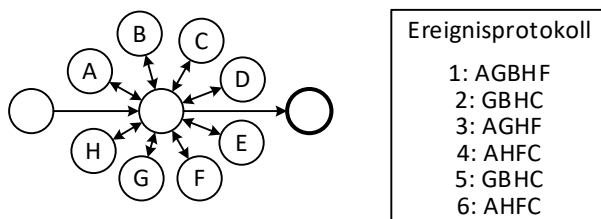


Abbildung 44: Unpräzises Modell für die Beschreibung eines Ereignisprotokolls

Die Bedeutung der Präzision ist für das adCM-Lösungskonzept als sehr hoch einzustufen, denn sie bildet gewissermaßen die Effizienz der Unterstützung des Case Managers ab: Wenn in einem vorgeschlagenen Template viele Agendaeinträge enthalten sind, die gar nicht im weiteren Verlauf des Falls benötigt werden, dann ist das Modell (in Form des Templates) nicht präzise und lenkt den Case Manager von seinen tatsächlich durchzuführenden Aktivitäten beziehungsweise benötigten Artefakten ab. Die Adaption und Anwendung der Metrik auf die Begriffe in dem adCM-Kontext ist eng angelehnt an die Definition von (Buijs et al. 2012). Darin wird die Anzahl ungenutzter ausgehender Kanten des Modells ins Verhältnis gesetzt zu der Anzahl an ausgehenden Kanten insgesamt. Die adaptierte Metrik ist so entworfen, dass die Anzahl der vorgeschlagenen und im Ereignisprotokoll beobachtbaren Templateelemente ins Verhältnis gesetzt wird zur Anzahl an vorgeschlagenen Templateelementen insgesamt.

Definition Präzision (Q_P): Sei V die Menge aller durch ein Template vorgeschlagenen Agendaeinträge und B die Menge aller im Protokoll beobachteten Ereignisse. Dann ist die Präzision wie folgt definiert:

Formel 7: Präzision im adCM-Kontext

$$Q_P = \frac{|V \cap B|}{|V|}$$

Die **Fitness** – gleichbedeutend mit dem Begriff „Recall“ aus dem Information Retrieval – beschreibt im Gegensatz zur Präzision nicht, wie gut das Modell zur Praxis passt, sondern umgekehrt, wie gut die Praxis zum Modell passt. Im Process Mining ist die Fitness demnach ein Maß dafür, wie viele der in der Praxis beobachteten Aktivitäten im Prozessmodell beschrieben wurden. Wenn beispielsweise viele Ereignisse im Ereignisprotokoll nicht durch das Modell erklärt werden können, hat das Modell eine geringe Fitness. Die maximale Fitness hingegen erreicht ein Modell nur dann, wenn jede protokollierte Aktivität im Modell abgebildet ist. Um die Abgrenzung dieser Metrik zu anderen Metriken zu verstehen, stelle man sich ein Modell vor, das mithilfe der wichtigsten Aktivitäten nur den „Happy Path“ eines Prozesses widerspiegelt. Am oben abgebildeten Protokoll ist das beispielsweise die Reduzierung des Modells auf die zwei häufigsten Traces (siehe Abbildung 45). Ein solches Modell ist unbestritten einfach und auch präzise – schließlich werden alle im Modell enthaltenen Elemente wegen ihrer Zugehörigkeit zum „Happy Path“ mit hoher Wahrscheinlichkeit auch in der Praxis beobachtet. Je nachdem wie extrem die Auswahl der wichtigsten Aktivitäten vorgenommen wurde, wird es aber mehr oder weniger viele Fälle geben, die vom Standardfall abweichen, folglich nicht zum Modell passen und deshalb die Fitness reduzieren.

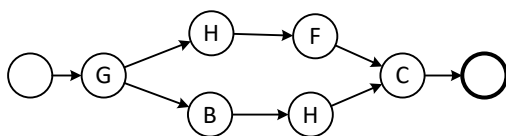


Abbildung 45: Präzises Modell mit schlechter Fitness bezogen auf das Protokoll aus Abbildung 44

Während die Präzision gewissermaßen die Effizienz des Lösungskonzepts widerspiegelt, so beschreibt die Fitness eher dessen Effektivität. Sie stellt im Kontext von schwach strukturierten Prozessen eine große Herausforderung bei der Implementierung von Discovery-Algorithmen dar. Denn die Fälle sind definitionsgemäß sehr unterschiedlich und aufgrund dieser breiten Streuung kann kaum ein repräsentativer Ablauf identifiziert werden. Mit einem kontrollflussorientierten Prozessmodell wäre die maximale Fitness im Kontext von adCM im Übrigen nur dann zu

erreichen, wenn man jeden Fall als überwiegend von anderen Pfaden isolierten Ablauf vom Start- bis zum Endknoten abbildet.

Im adCM-Lösungskonzept wird die Fitness in Anlehnung an (Buijs et al. 2012) so definiert, dass die Anzahl der vorgeschlagenen Templateelemente, die auch im Ereignisprotokoll beobachtet werden, ins Verhältnis gesetzt werden zur Anzahl aller Ereignisse im Protokoll.

Definition Fitness (Q_F): Sei V die Menge aller durch ein Template vorgeschlagenen Agendaeinträge und B die Menge aller im Protokoll beobachteten Ereignisse. Dann ist die Fitness wie folgt definiert:

Formel 8: Fitness im adCM-Kontext

$$Q_F = \frac{|V \cap B|}{|B|}$$

Abbildung 46 soll die **mengentheoretischen Zusammenhänge von Fitness und Präzision** und deren Abbildung auf die adCM-Konzepte an einem Venn-Diagramm veranschaulichen. Darin wird die Menge der vorgeschlagenen Template-Elemente (links) der Menge an beobachteten Ereignissen (rechts) gegenübergestellt.

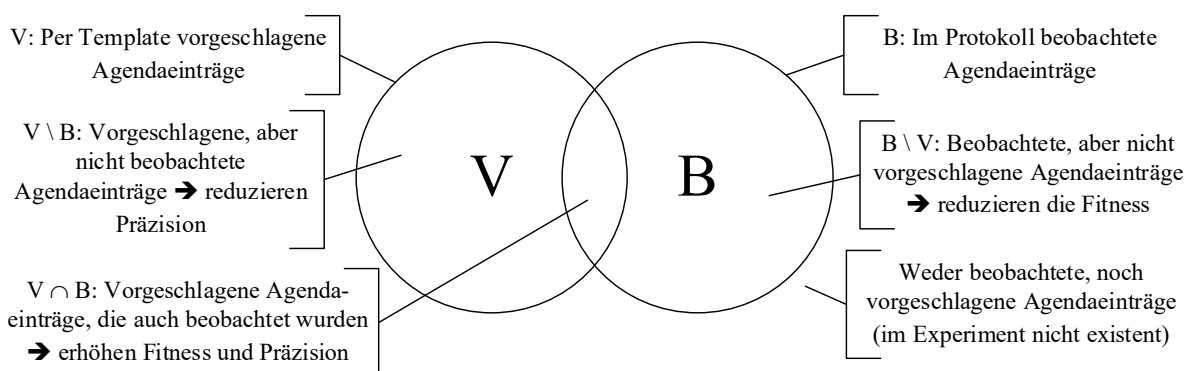


Abbildung 46: Venn-Diagramm zur Veranschaulichung der Zusammenhänge zwischen Fitness und Precision im Kontext von adCM

Aus zwei Gründen ist die Fitness zwar ein wichtiges, aber dennoch der Präzision **untergeordnetes Qualitätsziel** im Kontext von adCM. Erstens stellt die Unterstützung von schwach strukturierten Prozessen – im Gegensatz zu strukturierten Prozessen – bislang noch ein Forschungsdesiderat dar, sodass prinzipiell jeder Zugewinn an Fitness im Sinne *effektiver* Unterstützungsleistung bereits ein Erfolg ist. Zweitens ist die Präzision im Sinne der *Effizienz* des Lösungskonzepts die einzige Metrik, welche die unaufdringliche Unterstützung des Case Managers messen kann – einer der zentralen Anforderungen an das adCM.

Unter **Generalization** versteht man eine Metrik, die das Abstrahierungsvermögen eines Modells bemisst (van der Aalst 2011). Dieses Vermögen beschreibt die Fähigkeit, bislang unbekanntes – im Sinne von bislang untrainiertem – Verhalten aus Ereignisprotokollen erklären zu können. Es handelt sich also um eine Art zukunftsbezogene Fitness. Eine solche Metrik ist notwendig, weil in der Process Mining-Disziplin die Fitness üblicherweise retrospektiv zu dem Ereignisprotokoll gemessen wird, mit dem das Modell erzeugt wurde. Bezogen auf das oben genannte Beispiel würde ein schlecht abstrahierendes Modell nur genau diejenigen Traces zulassen, die bislang im Protokoll zu finden sind. Sobald ein Case Manager eine andere Abarbeitungsreihenfolge wählt oder eine neue Aktivität hinzukommt, kann das Modell diesen neuen Trace nicht beschreiben, was sich in einer reduzierten Fitness ausdrückt. In (van der Aalst 2011) werden die Enumeration Models als Beispiele für schlecht abstrahierende Prozessmodelle angeführt. Sie enthalten jeden möglichen Pfad genau einmal und müssten um einen ebensolchen Pfad ergänzt werden, sobald ein neues Verhalten durch das Modell erklärt werden muss (siehe Abbildung 47, links). Im Gegensatz dazu sind die bereits oben beschriebenen „Flower-Modelle“ Extrembeispiele für sehr gut generalisierende Modelle (siehe Abbildung 47, rechts), allerdings zulasten der Präzision.

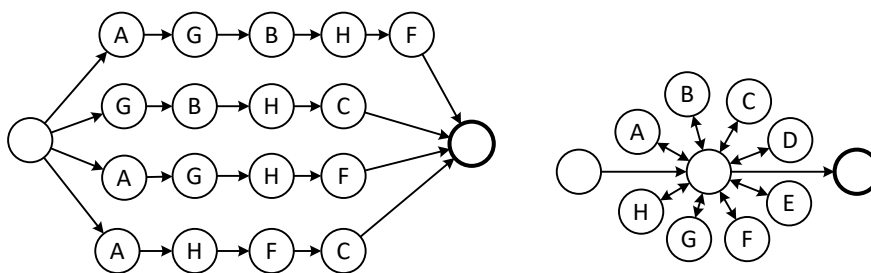


Abbildung 47: Modell mit schlechter (links) und guter (rechts) Abstraktion zur Beschreibung des Ereignisprotokolls aus Abbildung 44

Für einen Case Manager ist das Abstraktionsvermögen des ihn unterstützenden Modells sehr wichtig, denn die Fallbasis, aus der das Modell trainiert werden kann, ist gemäß der Definition von schwach strukturierten Prozessen in der Regel nie vollständig. Das heißt, jeder neue Fall enthält mit hoher Wahrscheinlichkeit eine bislang unbekannte Sequenz an Ereignissen, die durch das Modell abgebildet werden sollte. In (Buijs et al. 2012) wird eine Definition vorgestellt basierend auf der Annahme, dass häufig beobachtete Elemente auch in zukünftigen Ereignisprotokollen auftreten werden. Die Generalization wird deshalb genau dann hoch bewertet, wenn im Modell hauptsächlich diese Art von Ereignissen abgebildet wird. Enthält ein Modell umgekehrt viele Ereignisse, die nur sehr selten auftreten, dann abstrahiert es schlecht. Im Kontext von schwach strukturierten Prozessen ist die oben genannte Annahme nicht haltbar. Stattdessen wird im adCM-Lösungskonzept eine Definition von Generalization eingeführt, die näher an dem ursprünglichen Verständnis von (van der Aalst 2011) ist. Wenn die Fitness bezogen

auf ein komplett unbekanntes Ereignisprotokoll ebenso hoch ist wie bezogen auf das Ereignisprotokoll, mit dem das Modell erzeugt wurde, dann ist die Generalization hoch. Diese Definition folgt dem Prinzip der Kreuzvalidierung, bei dem die Qualität eines Modells nicht mithilfe des Trainingsdatensatzes, sondern eines separaten Testdatensatzes ermittelt wird.

Definition Generalization (Q_G): Sei L_{Test} das Test-Ereignisprotokoll und $L_{Training}$ das Trainings-Ereignisprotokoll, dann ist Generalization wie folgt definiert:

Formel 9: Generalization im adCM-Kontext

$$Q_G = \frac{Q_F(L_{Test})}{Q_F(L_{Training})}$$

Als letzte Qualitätsmetrik wird **Simplicity** vorgestellt. In Anlehnung an Ockhams Rasiermesser folgt es dem Grundgedanken, dass von allen ein Phänomen erklärenden Modellen dasjenige ausgewählt werden sollte, das am einfachsten ist beziehungsweise mit weniger Elementen auskommt. Es ist offensichtlich, dass diese Definition von Einfachheit in Konflikt steht zu Fitness, denn oft kann eine hohe Fitness nur durch sehr komplexe Modelle erreicht werden, die alle Eventualitäten enthalten. (Buijs et al. 2012) definieren Simplicity, indem sie die Anzahl an redundanten und fehlenden Modellelementen ins Verhältnis setzen zur Anzahl an Modellelementen und Ereignissen insgesamt. Demnach ist das Modell dann einfach, wenn jede Aktivität im Ereignisprotokoll genau einem Modellelement zugeordnet werden kann.

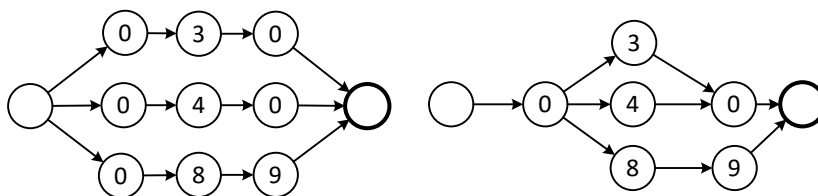


Abbildung 48: Ein kompliziertes (links) und ein einfaches (rechts) Modell für denselben Prozess

Aufgrund der sehr unterschiedlichen Konzepte eines einzelnen, schwergewichtigen Prozessmodells seitens Process Mining und einer Menge an unzusammenhängenden, leichtgewichtigen Templates, ist die Adaption dieser Definition von Einfachheit nicht praktikabel. Sofern nicht – wie in (Benner-Wickner et al. 2016) beschrieben – ein gezielter Ansatz zur Erzeugung und Pflege redundanzfreier Templates verfolgt wird, ist eine ungeordnete Menge von Templates prinzipbedingt nicht einfach. Trotzdem könnte man versuchen, die Größe des Modells mit dem zu erklärenden Verhalten in Beziehung zu setzen. Hierzu wird im adCM-Konzept die Annahme getroffen, dass die Größe des Modells verstanden werden kann als der Umfang der

gewonnenen Templates im Sinne der Summe der Elemente aller Templates. Dann könnte man die Simplicity errechnen, indem man diese Zahl ins Verhältnis setzt zur Anzahl an Aktivitäten im Ereignisprotokoll insgesamt.

Definition Simplicity (Q_s): Sei k_i die Anzahl der Elemente im i -ten Template und sei n die Anzahl der Templates insgesamt, dann ist die Simplicity wie folgt definiert:

Formel 10: Simplicity im adCM-Kontext

$$Q_s = \frac{\sum_{i=1}^n k_i}{\#Aktivitäten(L)}$$

6.4.1.2 Ereignisprotokolle auswählen

Möglichst geeignete Ereignisprotokolle für die Simulation zu wählen ist eine wichtige Aufgabe, weil alle danach folgenden Schritte auf dieser Entscheidung aufbauen. In erster Linie muss es sich um ein Protokoll eines Prozesses handeln, der den in dieser Dissertation fokussierten Prozessen möglichst ähnlich ist. Das heißt, es darf sich nicht um einen strukturierten Prozess handeln (Metriken zur Identifikation solcher Prozesse wurden in Kapitel 5.3.6 vorgestellt). Zweitens muss der Datensatz frei verfügbar sein, um den in (van der Aalst et al. 2012) formulierten Bedarf nach repräsentativen Benchmarks zu adressieren.

Die IEEE Task Force on Process Mining stellt mithilfe des gemeinsamen Daten-Repository der Universitäten Delft, Eindhoven und Twente eine Sammlung an Ereignisprotokollen zur Verfügung, mit denen Forscher beispielsweise ihre eigenen Process Mining-Algorithmen testen und mit anderen Ergebnissen vergleichen können⁵. Nur wenige dieser Ereignisprotokolle werden als „ad-hoc“-strukturiert bezeichnet, sodass sie gemäß der Klassifikation von (Loeffeler et al. 1998) in das Spektrum von denjenigen Prozessen fallen, die mit den adCM-Lösungskonzepten gezielt unterstützt werden sollen. Eines dieser Ereignisprotokolle stammt aus der Praxis und wurde in einem niederländischen Universitätsklinikum aufgezeichnet. Es handelt sich um dasselbe Protokoll, das bereits bei der Evaluation der Ereignisprotokoll-Komplexitätsmetriken verwendet wurde. Es enthält 1143 Fälle mit über 15000 Ereignissen und dient als Datenbasis für das in diesem Kapitel beschriebene Experiment zur Evaluation des adCM Miners.

Um die Belastbarkeit der in diesem Experiment gewonnenen Ergebnisse zu überprüfen, wird ein weiteres als „ad-hoc“-strukturiert bezeichnetes Ereignisprotokoll ausgewählt. Es enthält 832

⁵ https://data.3tu.nl/repository/collection:event_logs

Fälle mit über 44000 Aktivitäten aus einem Baugenehmigungsprozesses einer niederländischen Gemeinde (B.F. van Dongen 2015). Die Ergebnisse werden am Ende in Kapitel 6.4.3 gegenübergestellt.

6.4.1.3 Kreuzvalidierung vorbereiten

Wie bereits bei der Diskussion der Modellqualitätsmetriken angedeutet, ist es eine eher triviale Aufgabe, ein Modell mit einer perfekten Fitness zu gewinnen, obwohl es zugleich wahrscheinlich kein einfaches und präzises Modell wäre. Hierzu müsste das Modell nur perfekt auf das Ereignisprotokoll abgestimmt sein, das für die Erzeugung des Modells verwendet wurde. Es liegt nahe, dass ein speziell auf den Trainingsdatensatz zugeschnittenes Modell nicht notwendigerweise auch gut abstrahiert, also nicht unbedingt eine vergleichbare Fitness bei anderen Datensätzen aus demselben Prozess besitzt. Folglich berücksichtigt der Experimentaufbau eine Kreuzvalidierung, bei der das Ereignisprotokoll gemäß der im Data Mining üblichen Praxis in einen Trainings- und Testdatensatz aufgeteilt wird (van der Aalst 2011; Berry, Michael J. A. und Linoff 2000). Der Trainingsdatensatz wird verwendet, um das Modell zu erzeugen (siehe Abbildung 49). Bezogen auf den adCM-Aufbau handelt es sich dabei um eine Menge von Templates, während der Process Mining-Aufbau ein kontrollflussbasiertes Prozessmodell erstellt. Im Gegensatz dazu wird der Testdatensatz dazu verwendet, die darin enthaltenen Fälle mithilfe des jeweiligen Modells zu unterstützen. Beim adCM-Aufbau kommt hierzu notwendigerweise eine Vorschlagsfunktion zum Einsatz, während beim Process Mining-Aufbau die jeweils zum erzeugten Modelltyp passenden Replay-Algorithmen verwendet werden (siehe Abbildung 42).

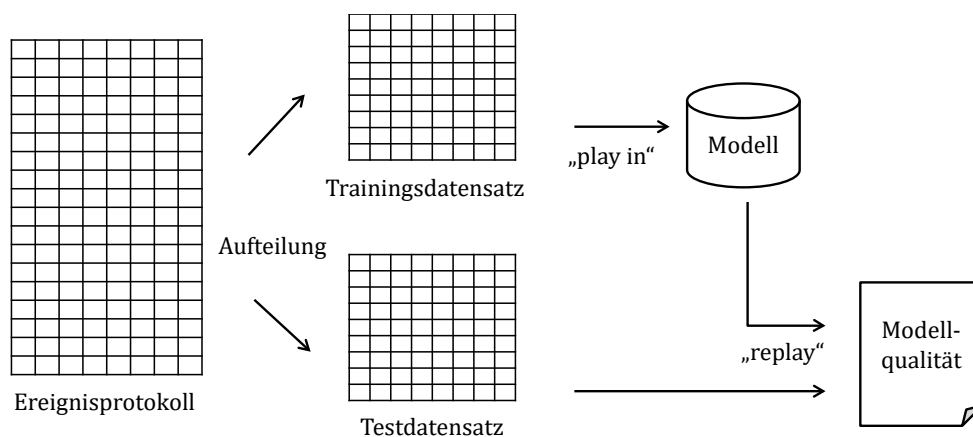


Abbildung 49: Anwendung der Kreuzvalidierung im Experimentaufbau

Weil es in der Literatur keine einheitliche Meinung dazu gibt, in welchem Verhältnis die Daten in Trainings- und Testdatensatz aufgeteilt werden sollen, wird das Ereignisprotokoll halbiert. Bei

ersten Testdurchläufen ist allerdings entdeckt worden, dass die erste Hälfte der Fälle eine höhere Diversität besitzt als die zweite Hälfte. Um Verfälschungen bei den Ergebnissen zu vermeiden, wurde das Ereignisprotokoll nicht in der Mitte geteilt. Stattdessen wurde jeder einzelne Fall mithilfe eines Zufallszahlengenerators willkürlich in einen der beiden Datensätze verschoben. Im Ergebnis enthielt das Ereignisprotokoll mit den Trainingsdaten 574 Fälle, während das Protokoll mit den Testdaten 569 Fälle enthielt.

6.4.1.4 Process Mining-Algorithmen auswählen

Der bereits weiter oben erwähnte Bedarf an repräsentativen Benchmarks erfordert nicht nur eine gut begründete Auswahl des Testdatensatzes, sondern auch eine sinnvolle Auswahl der Process Mining-Algorithmen, gegen die der adCM Miner „antreten“ soll. Hierzu müssen zunächst die Anforderungen an solch einen Algorithmus definiert werden. In diesem Zusammenhang ist es wichtig darauf hinzuweisen, dass es sich um die Unterstützung von schwach strukturierten Prozessen handelt, die wegen der hohen Diversität einen entsprechend hohen Anteil an selten vorkommenden Aktivitätsmustern enthält („Rauschen“). Aus diesem Grund müssen Algorithmen ausgewählt werden, die in der Literatur dafür ausgezeichnet wurden, mit rauschbehafteten Ereignisprotokollen umgehen zu können. Außerdem sollten die Algorithmen als Plug-In des Standard-Process Mining-Werkzeugs „ProM“ verfügbar sein. Das hat die folgenden Vorteile: Erstens sind die Algorithmen dadurch einfach anwendbar, weshalb das Experiment ohne umfassende zusätzliche Programmierarbeiten durchgeführt werden kann. Zweitens sind die Algorithmen dann auch in der Lage, die Ereignisdaten in ihrer vorliegenden, standardisierten Form (eXtensible Event Stream, XES) einzulesen. Auf der Grundlage dieser Anforderungen wurden die folgenden fünf Discovery-Algorithmen als Maßstab für das Experiment ausgewählt:

- ILP Miner (van der Werf et al. 2008)
- InductiveMiner (Leemans et al. 2013)
- Heuristics Miner (Weijters und de Medeiros 2006)
- Flexible Heuristics Miner (Weijters und Ribeiro)
- ETM (Buijs et al. 2012)

6.4.1.5 Agenda-Import konfigurieren

Es sei daran erinnert, dass der adCM Miner anstelle eines herkömmlichen Ereignisprotokolls eine Graphstruktur aus semantisch verknüpften, abgeschlossenen Fällen benötigt. Die Agenden liegen dabei definitionsgemäß in einer hierarchischen Struktur vor und nicht in einer Sequenz aus Aktivitäten. Um diesen Unterschied im Experiment zu berücksichtigen, musste ein Weg gefunden werden, die Fälle im Ereignisprotokoll derart in eine Agendastruktur zu transformieren, als wäre

diese Sequenz ein Resultat des Abarbeitens einer Agenda durch einen Case Manager. Zu diesem Zweck wurde eine Komponente entwickelt, die zu jedem Fall in den Ereignisprotokollen eine passende Agenda erzeugen kann. Dabei muss eingestanden werden, dass es zahlreiche Strategien geben kann, nach denen ein Case Manager eine Agenda strukturiert. Beispielsweise könnte er die Aktivitäten nach den zu erreichenden Meilensteinen, grundlegenden Domänenbegriffen oder zentraler Arbeitsergebnisse sortieren. Ebenfalls muss eingestanden werden, dass die erzeugten Agenden abhängig von der gewählten Strategie sehr unterschiedlich aussehen. Um diese Schwachpunkte zu untersuchen, wurden verschiedene Vorab-Simulationen durchgeführt, bevor das eigentliche Experiment startete.

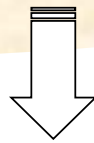
Diese Simulationen hatten zum Inhalt, verschiedene Strategien zu testen und hinsichtlich ihres Einflusses auf die Qualitätsmetriken zu untersuchen. Mögliche Strategien zur Strukturierung der Agenden ergaben sich durch eine Analyse der im Ereignisprotokoll enthaltenen Metadaten. Darin fanden sich einige vielversprechende Attribute, die nicht nur rein strukturell, sondern auch semantisch eine sinnvolle Aufteilung boten. Sie teilen die Aktivitäten in einem Fall in etwa gleich große Cluster auf, die einer nachvollziehbaren Logik folgen. Hierzu zählt das Attribut „org:group“, das auf die Organisationseinheit verweist, in der die Aktivität durchgeführt wurde (in diesem Fall die Station, in der die Behandlung stattgefunden hat). Es wäre nicht ungewöhnlich, wenn ein Case Manager diese Aufteilung vorgenommen hätte, um die Aktivitäten zu untergliedern – schließlich werden in den Stationen jeweils unterschiedliche Aktivitäten vorgenommen, die entlang des klinischen Behandlungspfades auch Meilensteine darstellen können, wie zum Beispiel das Verlegen eines Patienten von der Intensivstation in die Innere Medizin. Ein weiteres Attribut ist der „specialism code“, ein domänenspezifischer Begriff, der sich ähnlich gut zum Strukturieren der Aktivitäten eignet.

Es ist unbestritten, dass diese Transformation nie die individuellen, kontextbezogenen Strukturierungsentscheidungen treffen kann, die ein Case Manager mit seinem Expertenwissen abrufen kann. Deshalb ist es an dieser Stelle wichtig zu motivieren, dass die Art und Weise der Strukturierung keinen wesentlichen Einfluss auf die Qualität der Unterstützung und erst recht nicht auf die Machbarkeit hat. Untermauert wird diese Annahme durch die Vorab-Experimente mit den unterschiedlichen Strategien zur Erzeugung der Agenden: Die Qualitätsmetriken unterschieden sich im Mittel nur um 0,27%, mit einer Standardabweichung von 17,66%. Diese geringen Abweichungen können dadurch erklärt werden, dass der adCM Miner lediglich die Aufgabe hat, häufig wiederkehrende Muster in Agendabäumen zu identifizieren. Mit welcher Semantik diese Bäume erstellt wurden, ist bei dieser Aufgabe nicht entscheidend. Trotzdem musste nach den Vorab-Experimenten eine Entscheidung getroffen werden, welche Strategie für

die Evaluation ausgewählt wird. Weil das „org-group“-Attribut sowohl semantisch als auch hinsichtlich der Struktur die besten Eigenschaften zur Gruppierung der Aktivitäten besitzt, wurde die Transformation anhand der darin enthaltenen Informationen durchgeführt (siehe Abbildung 50).

Ereignisprotokoll:

```
<trace>
  <event>
    <string key="org:group" value="Obstetrics & Gynaecology clinic"/>
    <string key="concept:name" value="1e consult poliklinisch"/>
  </event>
  <event>
    ...
  </event>
</trace>
```



Transformation

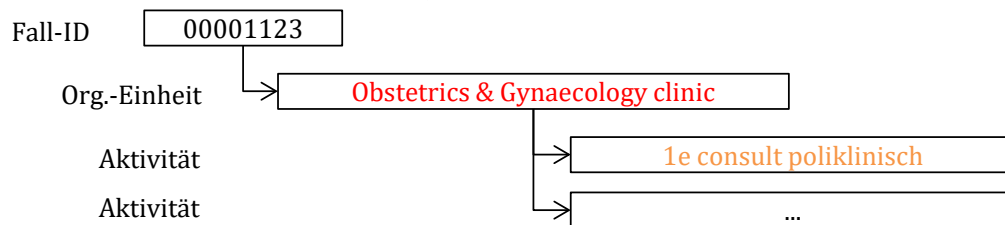


Abbildung 50: Transformation der Traces im Ereignisprotokoll in Agendastrukturen

6.4.1.6 Template-Auswahl konfigurieren

Der zweite und letzte Eingriff in die Konfiguration des Experiments bezieht sich auf die Auswahl der vorgeschlagenen Templates. Weil es sich um ein automatisiertes Experiment mit über 1000 zu verarbeitenden Fällen handelt, muss eine Komponente den Case Manager bei der Auswahl der vorgeschlagenen Templates ersetzen. Hierzu wurden analog zum Vorgehen bei der Erzeugung der Agenden verschiedene Strategien erarbeitet und in einem Vorab-Experiment hinsichtlich ihrer Auswirkungen auf die Qualitätsmetriken überprüft. Die Strategien verfolgen alle das Ziel, aus der Menge an vorgeschlagenen Templates ein Template auszuwählen und in den zum Zeitpunkt des Vorschlags aktuellen Stand der Agendastruktur zu integrieren.

Eine der Strategien zielte darauf ab, das Template auszuwählen, was am besten zu dem aktuellen Fallkontext passt. Hierzu wurde nach einem Abgleich aller vorgeschlagenen Templates mit der aktuellen Agenda genau das Template ausgewählt, das die größte Schnittmenge an Knoten besitzt. Diese Strategie nimmt an, dass eine solch große Schnittmenge auch ein Indiz dafür ist, dass die dann vorgeschlagenen, noch nicht in der Agenda vorhandenen Templateelemente besonders nützlich sind. In diesem Sinne verfolgt sie das Ziel, eine hohe Präzision zu erreichen (siehe Kapitel 6.4.1.1).

Eine weitere Strategie verfolgte den fast umgekehrten Weg und wählte aus den vorgeschlagenen Templates das größte aus. Diese Strategie ist auf eine hohe Fitness ausgelegt, weil durch die Integration großer Templates in die bestehende Agenda die Wahrscheinlichkeit steigt, dass die noch folgenden Aktivitäten dann bereits enthalten sind. Offensichtlich konkurriert diese Strategie mit der ersten Strategie, weil die Integration besonders großer Templates eben auch die Wahrscheinlichkeit erhöht, dass einige davon nicht benötigt werden.

Um eine Balance zwischen beiden Qualitätsmetriken zu schaffen, wurde ein hybrider Ansatz ausgewählt. Er sortiert unter den vorgeschlagenen Templates alle Templates aus, die eine minimale Übereinstimmung mit der aktuellen Agenda nicht erreichen und wählt aus dieser Menge das größte Template.

Das Ersetzen eines Case Managers durch eine automatische Komponente zur Auswahl der Templates ist in zweierlei Hinsicht problembehaftet: Erstens hat die Wahl des Templates einen unmittelbaren Einfluss auf die finale Struktur der Agenda. Zweitens ist es sehr schwierig, wenn nicht gar unmöglich, eine Heuristik zu entwickeln, die die Qualifikation und langjährige Erfahrung eines Case Managers bei der Template-Auswahl ersetzen kann. Aber gerade weil der Case Manager in den meisten Fällen sicherlich die besseren Entscheidungen treffen wird, kann man davon ausgehen, dass alle Ergebnisse, die aus den oben diskutierten Auswahlstrategien hervorgehen, immer nur eine Art untere Grenze der Ergebnisqualität darstellen. Zusammengefasst ist also das Ergebnis der Evaluation als minimal erreichbare Qualität zu verstehen, die durch einen einigermaßen erfahrenen Case Manager nur verbessert werden kann.

6.4.1.7 Experiment durchführen und bewerten

Nach den Schritten zur Vorbereitung des Experiments wurde mit der Evaluation des adCM Miners begonnen (obere Teststrecke auf Abbildung 42). Anschließend erst wurden zum Vergleich die Process Mining-Algorithmen angewendet (untere Teststrecke).

Die adCM-Werkzeugkette startete mit der Transformation des Ereignisprotokolls in einen Wald aus Agenden. Diese Agenden wurden gemäß dem Monitoring-Lösungskonzept semantisch mit den adCM-Konzepten verknüpft und in eine Graphdatenbank persistiert. Anschließend wurde der adCM Miner angestoßen, um aus den Agenden Templates zu extrahieren. Dabei wurden verschiedene Schwellwerte für den minimalen Support verwendet, um den Einfluss dieses Parameters auf die Ergebnisqualität zu untersuchen (siehe Kapitel 6.4.2.1). Daran angeknüpft fand das „abspielen“ des Testdatensatzes statt (replay). Hierzu musste der dafür abgetrennte Teil des Ereignisprotokolls ebenfalls in Agendabäume transformiert werden. Während die

Agendabäume durch inkrementelles Hinzufügen von Aktivitäten aus dem Testprotokoll „wuchsen“, wurde eine durch den Recommender ausgewählte Menge an mutmaßlich nützlichen Templates vorgeschlagen und von der Template-Auswahlkomponente in die Agenda integriert. Gemäß Abbildung 46 wurde durch eine Komponente zur Messung der Qualitätsmetriken mitgezählt, ob ein neuer Agendaeintrag a schon durch ein integriertes Templateelement abgedeckt war (also $a \in V \cap B$) oder nicht (also $a \in B \setminus V$). Sobald ein Fall gänzlich abgespielt wurde, konnte aus dem Testdatensatz jeweils ermittelt werden, welche Elemente in der finalen Agenda zwar aus einem Template stammen, jedoch nicht während des Abspielens beobachtet wurden (also $a \in V \setminus B$). Anhand dieser Informationen konnten dann die Werte für die Fitness und Präzision errechnet werden. Dieser Replay-Algorithmus ist auch in Listing 9 beschrieben. Für Details zur konkreten Arbeitsweise des Recommenders – also wie die Menge an vorgeschlagenen Templates zustande kommt – wird auf Kapitel 7.5 verwiesen.

```

log ← loadXESLog();
setStrategies();
FOR (each trace in log)
  agenda ← new Agenda();
  FOR (each event in trace)
    IF (event is new)
      item ← new AgendaItem(event.name);
      agenda.add(item, transformationStrategy);
    ELSE
      item ← getAgendaItem(event.name);
      item.isReplayed ← true;
    ENDIF
  templates ← recommender.getMatchingTemplates(agenda);
  IF (templates.size > 0)
    agenda.add(templates, applicationStrategy);
  END IF;
END FOR;
measureQuality(agenda);
END FOR;

```

Listing 9: Replay-Algorithmus der adCM-Teststrecke

Nachdem alle Fälle aus dem Testdatensatz abgespielt und alle Metriken ermittelt wurden, sind die verschiedenen Discovery- und Replay-Algorithmen aus dem Process Mining auf die Datensätze angewendet worden. Die Ergebnisse beider Ansätze werden im nachfolgenden Kapitel beschrieben.

6.4.2 ERGEBNISSE

In diesem Kapitel werden die Ergebnisse des Experiments erläutert. Das Kapitel beginnt mit den Zwischenergebnissen aus dem Discovery-Schritt, gefolgt von den Ergebnissen des Replay-

Schritts. In diesen Unterkapiteln wird jeweils zuerst das Ergebnis der Process Mining-Teststrecke geschildert, dann das Ergebnis der adCM-Teststrecke.

6.4.2.1 Discovery

Schon in diesem ersten Schritt hat sich gezeigt, dass nicht alle Process Mining-Algorithmen überhaupt in der Lage sind, mit der Komplexität des Ereignisprotokolls zurechtzukommen⁶. Nachdem sowohl der ILP- als auch der ETM-Algorithmus trotz mehrfacher Anläufe und unter unterschiedlichen Parametern auch nach mehreren Tagen noch nicht terminierten, beziehungsweise den zur Verfügung stehenden Arbeitsspeicher vollständig aufzehrten, wurden die Versuche ergebnislos abgebrochen. Beide Algorithmen konnten daher nicht weiter in die Ergebnisbetrachtung aufgenommen werden. Im Gegensatz dazu haben der Inductive Miner, der Heuristics Miner und der Flexible Heuristics Miner erfolgreich Prozessmodelle erzeugt. Die Modelle enthielten jeweils mehr als 600 Knoten.

Bezogen auf die adCM-Teststrecke zielt der Discovery-Schritt auf das erste Ziel des Experiments ab, nämlich die grundsätzliche Funktionstüchtigkeit des in Kapitel 0 konzipierten adCM Miners zu überprüfen. Zu diesem Zweck wurde der adCM Miner mit verschiedenen Support-Schwellwerten getestet, des einzigen Parameters für den Algorithmus. Die Ergebnisse (siehe Abbildung 51) zeigen, dass die Anzahl an gewonnenen Templates (Y-Achse) mit Steigerung des Support-Schwellwertes (X-Achse) sinkt. Dieser Zusammenhang ist vergleichbar mit dem Zusammenhang zwischen Qualität (wenige, aber dafür wertvolle Templates) und Quantität (viele, aber weniger wertvolle Templates): Sobald der Schwellwert – und damit der Anspruch an die Häufigkeit der identifizierten Muster – angehoben wird, können tendenziell weniger Templates gefunden werden. Also angenommen, bei der initialen Konfiguration eines adCM-Werkzeugs wird die Entscheidung getroffen, dass nur Templates erzeugt werden sollen, die mindestens einen Schwellwert von 60% erreichen. Dann werden bei dem hier untersuchten Datensatz zwar nur 10 Templates gefunden, die aber immerhin in weit mehr als der Hälfte der Fälle tatsächlich benötigt worden wären. Wird hingegen ein etwas niedrigerer Schwellwert in Höhe von 44% gewählt, dann werden bereits über 400 Templates entdeckt.

⁶ Diese Beobachtung deckt sich auch mit Erkenntnissen aus einer studentischen Abschlussarbeit (Akar 2014).

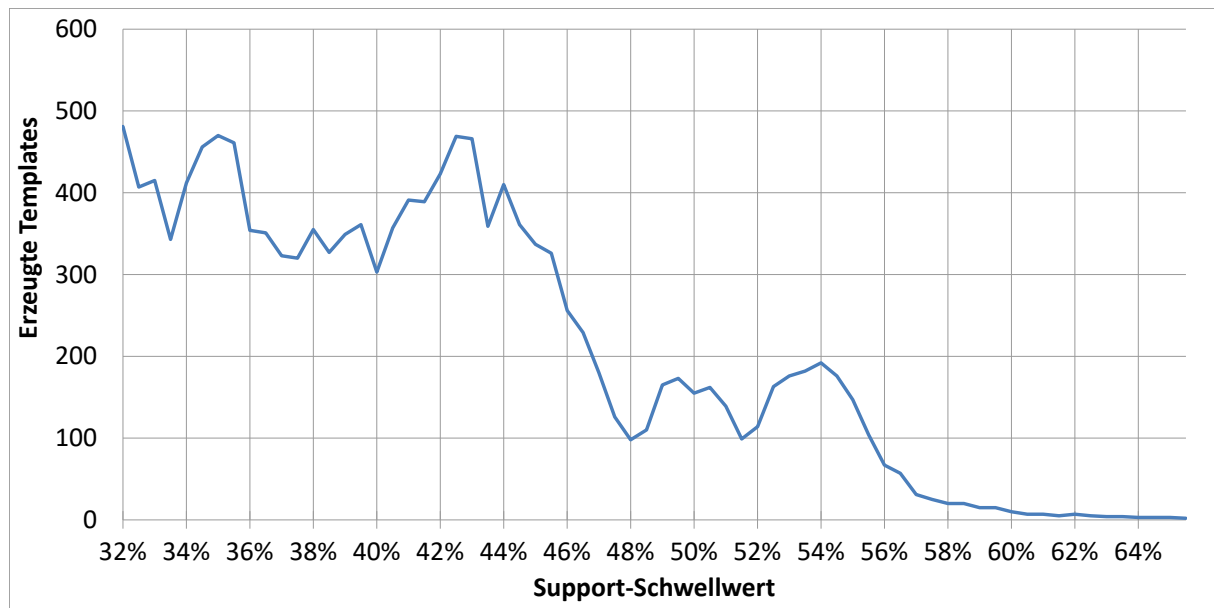


Abbildung 51: Auswirkungen von Schwellwertänderungen auf die Anzahl erzeugter Templates

Obwohl dieser Zusammenhang zwischen Qualität und Quantität grundsätzlich im abgebildeten Graph erkennbar ist, so scheint es dennoch lokale Abweichungen von einer mathematisch idealen Trendlinie zu geben. Beispiele hierfür sind die Maxima bei den Schwellwerten 43% und 54% beziehungsweise die Minima bei 40%, 48% und 51%. Dieses Phänomen kann durch die grundsätzliche Arbeitsweise des Algorithmus erklärt werden, die je nach Beschaffenheit der Daten an bestimmten Stellen des Ergebnisses zum Vorschein kommt: Algorithmen wie der adCM Miner, die gezielt nach *maximalen* Teilbäumen suchen, sind so konstruiert, dass sie Bäume aus dem Wald streichen, obwohl diese Bäume den Schwellwert erreichen. Das passiert immer dann, wenn die Bäume schon in Form von Teilbäumen in dem Wald enthalten sind. Je nach Verteilung der Häufigkeiten einzelner Muster in der Agenda kann es also mal häufiger und mal seltener vorkommen, dass durch die Reduktion des Schwellwertes viele kleine, sehr häufig auftretende Teilbäume in wenigen maximalen Bäumen aufgehen, die erst durch den geringeren Schwellwert Teil des Ergebnisses geworden sind. Abbildung 52 zeigt diesen Zusammenhang an einem Beispiel. Hier wurden bei einem Support-Schwellwert von 39% weniger Templates gefunden als bei 40%, weil die kleinen, häufigen Templates Teilbäume eines größeren Baumes sind. Er konnte erst durch die Reduktion des Schwellwertes entstehen, weil durch die Hinzunahme von „W“ der gesamte Baum unter die 40%-Schwelle fällt.

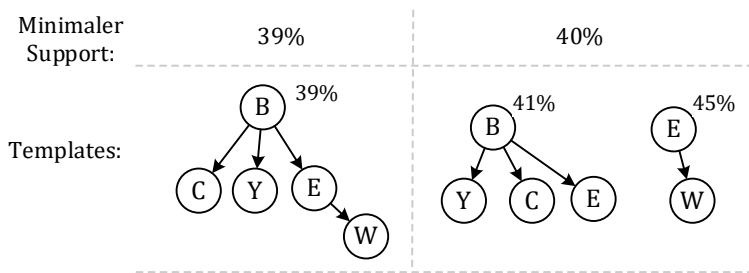


Abbildung 52: Beispiel für das "Verschlucken" von kleinen Teilbäumen durch größere, seltenere Bäume

Doch um die Auswirkungen dieser Phänomene auf die Qualität der Ergebnisse untersuchen zu können, wurde nicht einfach das Mining-Ergebnis zu einem willkürlich gewählten Schwellwert weiterverarbeitet, sondern zu drei unterschiedlichen, möglichst charakteristischen Schwellwerten. Aufgrund des oben beschriebenen Zusammenhangs zwischen Qualität und Quantität sind hierfür sicherlich die lokalen Maxima aus Abbildung 51 besonders interessant – schließlich weisen sie darauf hin, wo zu einem bestimmten Schwellwert der adCM Miner besonders ertragreich ist. Um diesen zunächst hypothetischen Zusammenhang zu prüfen, wurde aber auch ein Minimum ausgewählt. Als Ergebnis dieser Überlegungen wurde der nachfolgende Replay-Schritt mit drei Template-Mengen durchgeführt, die jeweils einen Support von 42,5 %, 51,5 % und 54 % besitzen.

Von dem oben beschriebenen Phänomen unberührt lässt sich abschließend feststellen, dass der adCM Miner fähig ist, Templates aus der Datenbank zu erzeugen (ein Beispiel-Template ist in Anhang 10.5 zu finden). Somit ist das erste Ziel des Experiments also bereits erreicht. Es kann aber nach wie vor noch keine Aussage darüber getroffen werden, ob diese Templates auch tatsächlich nützlich sind beziehungsweise eine hohe Qualität haben. Das Erreichen dieses Ziels lässt sich erst nach Abschluss des Replays bewerten. Dessen Ergebnisse werden im nächsten Abschnitt beschrieben.

6.4.2.2 Replay

Von den ursprünglich fünf Process Mining-Algorithmen haben im Discovery-Schritt jeweils nur der InductiveMiner, der Heuristics Miner und der Flexible Heuristics Miner ein Prozessmodell erstellen können. Sie sind demnach die einzigen verbliebenen Algorithmen, deren Ergebnis potentiell im Replay-Schritt weiterverarbeitet werden konnte. Doch auch bei diesem Schritt hat sich gezeigt, dass nicht alle Algorithmen auf schwach strukturierte Prozesse angewendet werden können, denn manche Modelle waren zu komplex, als dass sie mit dem vom Ereignisprotokoll abgespaltenen Testdatensatz abgeglichen werden konnten. Das betrifft erstens die Modelle des Flexible Heuristics Miner. Keine der verfügbaren Replay-Algorithmen konnten den Testdatensatz mit diesen Modellen so nachspielen, dass verlässliche Qualitätsmetriken daraus errechnet

werden konnten. Ebenso erfolglos war es, die Modelle des Heuristics Miner direkt zu überprüfen, denn wie sich herausstellte gibt es gar keine Algorithmen, die diese speziellen Modelle (Heuristics Net) verarbeiten können. Um dieses Problem zu umgehen, wurden – wie in (Lichtenegger 2012) beschrieben – spezielle Transformationswerkzeuge eingesetzt, mit denen sich die Originalmodelle zum Beispiel in Petrinetze umwandeln und im Replay weiterverarbeiten lassen. Zuletzt blieben also nur noch zwei Modelle für das Replay übrig: das eben genannte Petrinetz sowie das vom InductiveMiner erzeugte Petrinetz. Die Ergebnisse werden zusammen mit denen aus der adCM-Teststrecke diskutiert. Dort war der Replay mit allen drei ausgewählten Template-Wäldern erfolgreich. Die Tabelle 16 zeigt die errechneten Qualitätsmetriken sowohl für die gesamte adCM-Teststrecke (oben) als auch für die Process Mining-Teststrecke (unten). Neben den Qualitätsmetriken zeigt es auch die zur Messung der Simplicity relevante Anzahl an Modellelementen.

Tabelle 16: Gegenüberstellung der Qualitätsmetriken auf der Grundlage von insgesamt 1143 Fällen

<i>Algorithmus</i>	<i>Parameter</i>	<i>#Elemente</i>	<i>Precision</i>	<i>Fitness</i>	<i>Generalization</i>	<i>Simplicity</i>
adCM Miner	Support: 42,50%	7.909	61,56%	30,70%	92,21%	89,76%
	Support: 51,50%	1.146	69,12%	26,85%	99,29%	98,52%
	Support: 54,00%	4.816	71,48%	21,39%	100,65%	93,76%
Inductive Miner	Noise-Filter: 0%	621	6,43%	100,00%	100,00%	99,19%
	Noise-Filter: 20%			81,63%		
Heuristics Miner*	N/A	980	33,33%	56,20%	N/A	98,73%

*) Die Ergebnisse beziehen sich auf das aus dem ursprünglichen Modell erzeugte Petrinetz.

Gegenüberstellung der Präzision und der Fitness

Je nachdem, welchen Template-Wald man betrachtet, liegt die Präzision des adCM Miners im Intervall zwischen 61,56% und 71,48%. Daraus lässt sich ableiten, dass die überwiegende Mehrzahl der vorgeschlagenen Template-Elemente tatsächlich nützlich war, also im weiteren Verlauf des jeweiligen Falles auch als Aktivität auftrat. Vergleicht man die unterschiedlichen Precision-Werte der adCM Miners, so unterstützen die Ergebnisse die weiter oben beschriebene Annahme, dass ein hoher Support eines Templates auch einen positiven Effekt auf die Präzision hat. An dieser Stelle sei daran erinnert, dass der Case Manager aufgrund seiner Erfahrung bessere Entscheidungen bei der Auswahl der vorgeschlagenen Templates treffen kann als die recht einfache Vorschlagskomponente und dass die hier aufgeführten Ergebnisse somit nur eine untere Schranke der Präzision darstellen. Trotzdem konnten diese Ergebnisse von keinem der Process Mining-Algorithmen erreicht werden. Das liegt vor allem daran, dass diese Algorithmen nur dann einen schwach strukturierten Prozess unterstützen können, wenn sie es schaffen, ein sehr umfassendes Modell zu erzeugen. Wird ein konkreter Fall mithilfe eines solch großen Modells

unterstützt beziehungsweise „nachgespielt“, so sind die meisten darin enthaltenen Pfade irrelevant, was die Präzision drückt.

Wie bereits in Kapitel 6.4.1.1 diskutiert ist die Präzision eine wesentliche Qualitätseigenschaft, um den Case Manager auf eine möglichst unaufdringliche Art und Weise zu unterstützen – nämlich nur dann Templates vorzuschlagen, wenn sie mit besonders hoher Wahrscheinlichkeit benötigt werden. Leider lässt sich die Präzision oftmals nur zum Nachteil der Fitness optimieren (und umgekehrt, wie eben am Beispiel der Process Mining-Algorithmen beobachtet werden konnte). Denn wenn der Umfang der Templates zugunsten der Präzision reduziert wird, steigt zugleich das Risiko, dass manche Aktivitäten nicht mehr in den Templates enthalten sind. Dieses Phänomen lässt sich an den Ergebnissen des adCM Miners gut ablesen, denn die Fitness sinkt bei einer Precision von 71,48 % auf nur noch 21,39 %. Das bedeutet, dass nur 21,39 % der Aktivitäten durch die Templates vorhergesagt werden konnten. Verglichen mit den Replay-Ergebnissen, die mit den Modellen der Process Mining-Algorithmen erzielt wurden, ist die Fitness also sehr gering. Wegen des höheren Stellenwerts der Präzision im Sinne der Unterstützungseffizienz, wird die geringe Fitness als akzeptabel bewertet.

Generalization und Simplicity

Beim Replay hat sich herausgestellt, dass alle Modelle gut vom Trainingsdatensatz abstrahieren können unabhängig davon, ob es sich um Ergebnisse des adCM Miners oder des Inductive Miners handelt. Das heißt, die Fitness der Modelle hat sich beim Testdatensatz kaum verändert bezogen auf den Trainingsdatensatz. Es konnte allerdings keine erfolgreiche Messung für das Petrinetz durchgeführt werden, welches aus dem Heuristic Net erzeugt wurde. Alle dafür verfügbaren Algorithmen haben bei den entsprechenden Versuchen entweder Fehler produziert oder nicht terminiert. Eine detaillierte Untersuchung der Ergebnisse des adCM Miners deutet auf einen Zusammenhang zwischen dem minimalen Support der gewonnenen Templates und deren Abstraktionsvermögen hin: Je höher der Schwellwert beim Discovery eingestellt wurde, desto stärker scheinen die Templates auch auf zukünftige Fälle zu passen. Das geht in einer Konstellation sogar soweit, dass die Fitness für den Testdatensatz (21,53 %) minimal höher ist als für den Trainingsdatensatz (21,39 %), was definitionsbedingt zu einem eher ungewöhnlich zu interpretierenden Abstraktionsvermögen von 100,65% führt.

In Abhängigkeit vom gewählten Support-Schwellwert erzeugt der adCM Miner Hunderte von Templates, die bis zu 7900 Elemente enthalten. Diese Menge ist ganz offensichtlich nicht „einfach“ im Sinne eines menschenlesbaren Modells. Die hier ausgewählte Definition von Simplicity setzt die Modellgröße ins Verhältnis zu dem zu erklärenden Verhalten – hier repräsentiert durch die

Anzahl an Aktivitäten. Berücksichtigt man also, dass der Testdatensatz 77.238 Ereignisse enthält, sind die Modelle tatsächlich vergleichsweise einfach. Das trifft insbesondere natürlich auf den Template-Wald zu, der mit einem Support von 51,50 % erhoben wurde und aus nur einem Bruchteil an Templates besteht, verglichen mit den beiden anderen Template-Wäldern. Interessant ist in diesem Zusammenhang auch, dass die deutlich reduzierte Zahl an Templates keinen negativen Einfluss auf die Präzision hat. Das deckt sich mit den Erkenntnissen aus dem Discovery-Schritt.

Verglichen mit den Prozessmodellen des Heuristic Miners und des Inductive Miners haben die Template-Wälder eine schlechter ausgeprägte Simplicity. Dennoch enthalten die Prozessmodelle mehrere Hundert Elemente und sind somit ebenfalls nicht mehr menschenlesbar. Zusammengefasst stellt die Modellgröße aber nur dann ein Problem dar, wenn dadurch das Modell nicht mehr wartbar ist. In (Benner-Wickner et al. 2016) wird ein Ansatz vorgestellt, mit dem das Management von stark redundanten Templates vereinfacht werden kann.

6.4.3 EVALUATION MIT EINEM WEITEREN DATENSATZ

Um die oben beschriebenen Ergebnisse zu validieren, ist der gesamte Aufbau des Experiments im Rahmen einer studentischen Abschlussarbeit (Nisch 2016) mit einem weiteren Datensatz wiederholt worden. Dessen Eckdaten wurden bereits kurz in Kapitel 6.4.1.2 vorgestellt. Um die Agenda-Einträge zu strukturieren, wurde das Ereignis-Attribut „concept:name“ verwendet. Dessen Wert besteht jeweils aus drei mit einem Unterstrich getrennten Abschnitten, die eine fachliche Klassifikation des Ereignisses widerspiegeln.

Die Ergebnisse zeichnen ein ähnliches Bild: Beim Discovery werden je nach Schwellwert bis zu 400 Templates gefunden. Die Präzision liegt zwischen 74 % und 76 % und ist damit geringfügig höher als bei dem ersten Ereignisprotokoll. Die Fitness ist in etwa gleich hoch und liegt zwischen 20 % und 26 %. Auch bei dem zweiten Ereignisprotokoll ist es nicht gelungen, alle Process Mining-Algorithmen anzuwenden.

Abschließend kann anhand dieser Ergebnisse festgestellt werden, dass die Aussagen aus dem vorangegangenen Experiment valide sind.

6.5 ZUSAMMENFASSUNG

Das Discovery ist ein Teilbereich des Process Mining, der sich gezielt der Erzeugung von Prozessmodellen aus Ereignisprotokollen widmet. Im Kontext von adCM bedeutet das die

Erzeugung von Templates auf der Grundlage von Agenden abgeschlossener Fälle. Bevor aber ein passender Algorithmus für diese Aufgabe entwickeln werden konnte, musste zunächst die Beschaffenheit des dafür erforderlichen Datenmodells geklärt werden. Hierbei wurde herausgearbeitet, inwiefern die Adaption der Konzepte aus dem Semantic Web nutzbringend ist, um die von dem adCM-Konzept eingeforderte Flexibilität zu erreichen. Entlang des Linked Data-Prinzips und der Open World-Assumption wurde deshalb ergänzend zu dem Ergebnisbeitrag des Monitoring-Kapitels ein mehrschichtiges Datenmodell aufgebaut, das die Anforderungen aus dem Discovery adressiert.

Im Anschluss wurden die adCM-Lösungsbausteine zur Erzeugung von Templates beschrieben. Für die manuelle Definition von Templates wurde hierzu das „Teilen“-Konzept adaptiert, das aus dem Kontext mobiler Anwendungen bekannt ist. Einen aus dem wissenschaftlichen Standpunkt heraus weitaus interessanteren Aspekt stellen hingegen Algorithmen zur automatischen Erzeugung von Templates dar. Hierzu wurde der adCM Miner vorgestellt. Er basiert auf dem CMTreeMiner von (Chi et al. 2004) und ist mithilfe verschiedener Vor- und Nachbearbeitungsschritte in den adCM-Kontext integriert worden. Zu diesem Zweck beginnt der Algorithmus mit einem Clustering der verwendeten Begriffe auf Grundlage der im Datenmodell vorgehaltenen domänenspezifischen Vokabulare. Anschließend interpretiert er die Fallbasis als Wald von Agendabäumen und sucht darin nach maximalen Teilbäumen. Diese wiederum werden als Ergebnis in Form eines Template-Waldes ausgegeben und die geclusterten Agendaeintragsnamen werden mithilfe einer Heuristik wieder rücktransformiert.

Um die grundsätzliche Funktionstüchtigkeit und Nützlichkeit des Algorithmus zu evaluieren, wurde ein spezielles Experiment durchgeführt. Dessen Aufbau bestand aus zwei Teststrecken – je eine für das adCM-Lösungskonzept und für vergleichbare Algorithmen aus der Process Mining-Disziplin. Beide Teststrecken waren in zwei wesentliche Schritte unterteilt: dem Discovery und dem Replay. Im Discovery-Schritt wurden die Modelle in Form von Template-Wäldern einerseits und kontrollflussorientierten Prozessmodellen andererseits erzeugt. Dieser Schritt diente der Überprüfung der grundsätzlichen Funktionstüchtigkeit des adCM Miners. Weil unabhängig von den gewählten Parametern eine Vielzahl an Templates erzeugt werden konnte, ist dieses vorläufige Evaluationsziel erreicht worden. Im Gegensatz dazu waren nur wenige Process Mining-Algorithmen in der Lage, trotz der Komplexität des Ereignisprotokolls ein Prozessmodell zu erzeugen.

Um die Nützlichkeit der vom adCM Miner erzeugten Templates zu evaluieren, wurden mithilfe eines Replay-Schrittes Metriken erhoben. Es handelt sich dabei um Metriken, die im Process

Mining für die Bewertung von Discovery-Algorithmen allgemein anerkannt sind: Fitness, Precision, Generalization und Simplicity. Bezogen auf die Nützlichkeit sind insbesondere Fitness und Precision ausschlaggebend, denn sie können als die Effizienz und Effektivität der Modellunterstützung betrachtet werden. Die Auswertung der Ergebnisse hat bestätigt, dass nur wenige Process Mining-Algorithmen überhaupt in der Lage sind, überprüfbare Modelle für schwach strukturierte Prozesse zu erzeugen. Und selbst diese Algorithmen konnten nicht die Präzision des adCM Miners erreichen, dessen vorgeschlagene Template-Elemente je nach Konfiguration nur in 29-39 % der Fälle überflüssig waren. Auf der anderen Seite konnte mithilfe eines Process Mining-Algorithmus zwar ein Petrinetz erzeugt werden, welches jede Aktivität erklären konnte (100% Fitness), das aber wiederum so komplex war, nur ein sehr geringer Anteil der möglichen Pfade tatsächlich relevant war (6 % Precision).

Zusammengefasst lässt sich schlussfolgern, dass der adCM Miner für die Unterstützung schwach strukturierter Prozesse eine nützliche Alternative zu den Process Mining-Algorithmen darstellt. Letztere zeigten sich in der praktischen Anwendung mit dem Umfang und der Komplexität der Ereignisprotokolle überwiegend als ungeeignet. Insbesondere aufgrund der vergleichsweise sehr hohen Effizienz der vorgeschlagenen Templates zeigen die Ergebnisse, dass der adCM Miner das Paradigma einer unaufdringlichen Unterstützung erfüllt. Die Ergebnisse aus einem Vergleichsexperiment mit einem zweiten Ereignisprotokoll stützen diese Aussagen.

7 ARCHITEKTUR UND IMPLEMENTIERUNG DES ADCM-WERKZEUGS

In Kapitel 5 wurde erläutert, wie die Aktivitäten eines Case Managers aufgezeichnet werden können, damit sie auf der richtigen Detaillierungsebene in Form eines semantisch verknüpften Ereignisprotokolls festgehalten und ausgewertet werden können. Außerdem wurde ein Lösungskonzept beschrieben, wie der Kontext eines Falls anhand der in der Agenda befindlichen Informationen ermittelt werden kann. In Kapitel 6 wurde ein Datenmodell vorgestellt, das die aus dem Monitoring stammenden Informationen um zusätzliche Informationsebenen erweitert und somit als Grundlage für den anschließend beschriebenen Template Discovery-Algorithmus dient. Um zu zeigen, wie diese einzelnen Lösungsbausteine zu einem Werkzeug kombiniert werden können und über welche Schnittstellen sich dieses Werkzeug in eine Unternehmens-IT integrieren lässt, bedarf es eines Integrationskonzepts in Form einer **übergeordneten softwaretechnischen Architektur**.

Aufbauend auf diese Architektur werden in diesem Kapitel die **Design-Entscheidungen und Implementierungsdetails** zur Umsetzung der beiden zentralen Lösungsbausteine im adCM-Werkzeug motiviert und beschrieben. Weil sich die Monitoring- und Discovery-Funktionen hauptsächlich auf die Unterstützung der Kernaktivität „planen“ konzentrieren, werden in diesem Kapitel darüber hinaus Details über das Design und die Implementierung von zusätzlichen Unterstützungsfunktionen beschrieben. Dazu zählt eine Funktion zum Vorschlagen von Templates sowie weitere Funktionen, die auf die Kernaktivitäten „recherchieren“, „strukturieren“ und „interpretieren“ abzielen. Sie schließen den Zyklus zur Unterstützung des Case Managers (siehe Abbildung 53).

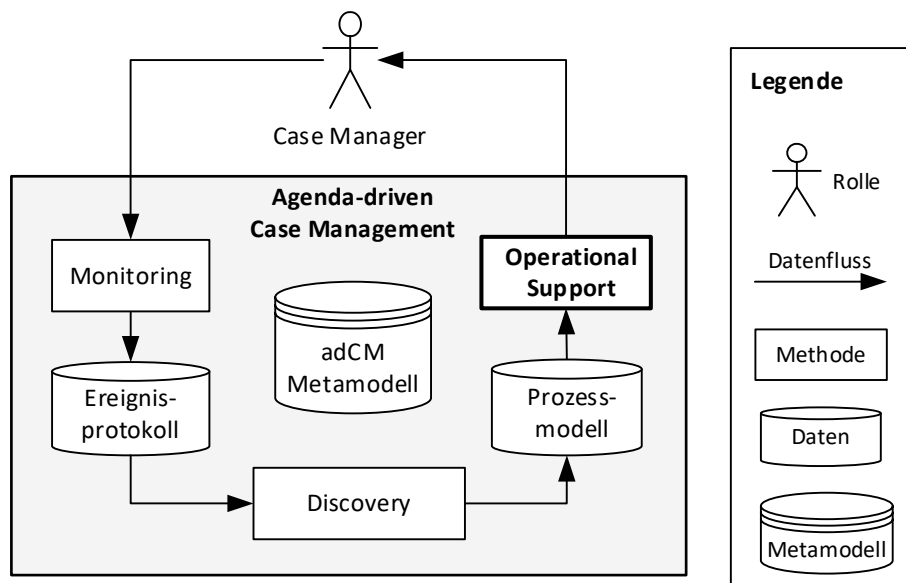


Abbildung 53: Die Unterstützungsfunktionen im adCM-Gesamtkontext

Das Kapitel beginnt zunächst mit einer kurzen Erläuterung der grundsätzlichen Vorgehensweise bei der Architekturgestaltung und einer Festlegung der dabei verwendeten Begriffe. Dann werden die Entwurfsentscheidungen zur übergeordneten softwaretechnischen Architektur beschrieben, gefolgt von den Details über das Design und die Implementierung des prototypischen adCM-Werkzeugs. Es schließt mit einer Zusammenfassung von Beiträgen aus der Literatur, die sich mit der Architektur von Werkzeugen vergleichbarer Ansätze beschäftigen.

7.1 VORGEHENSWEISE UND BEGRIFFSKLÄRUNG

Zur Definition der wesentlichen Komponenten einer Architektur wird in der Literatur die Berücksichtigung von etablierten Architekturmustern empfohlen (Rozanski und Woods 2012, S. 164). Auf der Grundlage von gesammelten Erfahrungen bieten sie Lösungsansätze für häufig wiederkehrende Probleme in der Architekturgestaltung. Nach (Vogel 2009, S. 205) handelt es sich bei Architekturmustern um Strukturen, die im Gegensatz zu Entwurfsmustern eher systemweit wirken und nicht lokal, beispielsweise bezogen auf eine Systemkomponente. Kann ein Problem nicht durch ein einzelnes Muster gelöst werden, so ist es üblich, mehrere Muster zu kombinieren (Buschmann 1996, S. 17; Rozanski und Woods 2012, S. 171; Starke 2015, S. 103). Anstelle des Begriffs Architekturmuster wird in der Literatur auch der Begriff Architekturstil verwendet, um Muster auf dieser Betrachtungsebene zu beschreiben (Rozanski und Woods 2012, S. 161). Im weiteren Verlauf wird durchgängig der Begriff Architekturmuster nach der Definition von Vogel verwendet.

Bezogen auf das Vorgehen zur Architekturgestaltung wird in der Literatur zudem der Einsatz von Szenarien nach (Bass et al. 2013) empfohlen, um architekturrelevante Anforderungen zu konkretisieren (Vogel 2009, S. 196; Starke 2015, S. 43; Rozanski und Woods 2012, S. 145).

Unter *Szenarien* sind Situationen zu verstehen, mit denen das System im Produktivbetrieb voraussichtlich konfrontiert werden wird. Sie werden mithilfe festgelegter Textschablonen so formuliert, dass sie zu einem gegebenen Auslöser (*Stimulus*), das gewünschte Systemverhalten oder die gewünschten Qualitätseigenschaften an Beispielen beziehungsweise konkreten Mengengerüsten verdeutlichen (*Antwort*). Damit ein solches Szenario präzise beschrieben ist, wird mit der *Umgebung* der Kontext oder Zustand angegeben, indem sich das System zum Zeitpunkt des Stimulus befindet. Außerdem wird jedem Szenario eine *Antwortmetrik* beigelegt, um die Bewertung der Szenarien zu unterstützen. Dabei kann es sich beispielsweise um eine Aussage handeln, die wahr sein muss, falls sich das System korrekt verhalten hat.

Anhand der Szenarien werden zu Beginn der Architekturgestaltung Architekturmuster identifiziert. Darüber hinaus dienen sie zur Bewertung bereits getroffener Architekturentscheidungen. Deshalb sollen Szenarien auch in diesem Kapitel eingesetzt werden.

7.2 ÜBERGEORDNETE SOFTWARETECHNISCHE ARCHITEKTUR

In diesem Abschnitt wird anhand einer softwaretechnischen Architektur beispielhaft gezeigt, wie die einzelnen Lösungsbausteine des Agenda-driven Case Management als System zusammengesetzt werden können und über welche Schnittstellen sich dieses System in eine Anwendungslandschaft integrieren lässt. Zur Motivation der getroffenen Entwurfsentscheidungen werden zunächst architekturrelevante Anforderungen und Qualitätseigenschaften anhand von Szenarien beleuchtet.

7.2.1 SZENARIEN FÜR ARCHITEKTURRELEVANTE ANFORDERUNGEN

Im Folgenden werden alle Anforderungen an die adCM-Konzepte aus Kapitel 4.1 bei der Entwicklung der Szenarios berücksichtigt unter der Annahme, dass sie alle grundsätzlich relevant für die Gestaltung der übergeordneten Architektur des Werkzeugs sind. Das Ergebnis ist in Anlehnung an das Schema aus (Bass et al. 2013) in der Tabelle 17 aufgeführt. Anschließend wird erörtert, inwiefern die Szenarien sich auf die Architekturgestaltung auswirken. Bezugnehmend darauf findet dann im nächsten Abschnitt die Auswahl der Architekturmuster statt.

Tabelle 17: Szenarien zur Konkretisierung architekturentwicklungsrelevanter Anforderungen nach (Bass et al. 2013)

Szenario-ID	1	Anforderungs-ID	1	Quelle	Case Manager
Auslöser	Der Case Manager beginnt einen Fall und möchte seinen gedanklich vorstrukturierten Plan externalisieren oder er setzt einen bereits begonnenen Fall fort und möchte seinen externalisierten Plan aufgrund einer veränderten Fallsituation neu strukturieren.				
Umgebung	Der Case Manager hat Zugriff auf das adCM-Werkzeug.				
Antwort	Das adCM-Werkzeug stellt dem Case Manager eine fallspezifische Agenda als Strukturierungshilfe zur Verfügung.				
Antwort-metrik	Die Agenda ist so flexibel umgestaltbar, dass deren Elemente jederzeit in ihrer Reihenfolge und in ihrer Verschachtelung verändert werden können.				
Szenario-ID	2	Anforderungs-ID	2	Quelle	Case Manager
Auslöser	Der Case Manager ergänzt seinen Plan um weitere Elemente, zum Beispiel neue Therapiemaßnahmen.				
Umgebung	Der Case Manager hat Zugriff auf das adCM-Werkzeug. Das Werkzeug hat Zugriff auf eine Fallbasis aus abgeschlossenen Fällen. Die Agenden der abgeschlossenen Fälle weisen Muster im Sinne häufig wiederkehrender Einträge auf. Die aktuelle Agenda des Case Managers stimmt teilweise mit den Elementen in den Mustern überein.				
Antwort	Das adCM-Werkzeug untersucht die Pläne in der Fallbasis auf wiederkehrende Elemente und speichert sie in Form von Templates. Es gleicht diese Templates mit der aktuellen Agenda des Case Managers ab und schlägt passende Templates vor.				
Antwort-metrik	Es werden alle Templates vorgeschlagen, die einen vorher festgelegten Ähnlichkeitswert beim Vergleich mit der aktuellen Agenda nicht unterschreiten.				
Szenario-ID	3	Anforderungs-ID	3	Quelle	Case Manager
Auslöser	Im Rahmen seiner Recherche arbeitet der Case Manager mit Informationen aus vielen unterschiedlichen Datenquellen (zum Beispiel Datenbanken, Repositories, Webseiten...)				
Umgebung	Das adCM-Werkzeug ist in die IT-Landschaft eines Unternehmens integriert. Es kann auf die Datenquellen des Unternehmens sowie auf das Internet zugreifen.				
Antwort	Das adCM-Werkzeug bietet dem Case Manager eine Suchfunktion an, mit der die angefragten Informationen gefunden werden können.				
Antwort-metrik	Das adCM-Werkzeug kann Suchergebnisse aus allen Quellen anbieten, an die es zuvor angeschlossen wurde.				
Szenario-ID	4	Anforderungs-ID	4	Quelle	Case Manager
Auslöser	Der Case Manager möchte ein (Zwischen-)Ergebnis seiner Recherche ablegen, zum Beispiel einen Link auf eine Intranetseite mit gesammelten Arztbewertungen.				

Umgebung	Der Case Manager hat Zugriff auf das adCM-Werkzeug und mindestens eine passende Information in den Datenquellen gefunden (zum Beispiel über die Suchfunktion)				
Antwort	Das adCM-Werkzeug bietet dem Case Manager einen Workspace an, auf dem er die Ergebnisse ablegen kann				
Antwort-metrik	Der Workspace kann alle Arten von Artefakten aufnehmen, die über die Suchfunktion erreichbar sind.				
Szenario-ID	5	Anforderungs-ID	5	Quelle	Case Manager
Auslöser	Der Case Manager möchte die Ergebnisse der Recherche strukturieren und miteinander in Beziehung setzen, zum Beispiel indem er eine Studie über die Wirksamkeit einer Therapiemaßnahme mit dem Heilverlaufsplan verknüpft.				
Umgebung	Der Case Manager hat Zugriff auf das adCM-Werkzeug und mindestens zwei Artefakte im Workspace abgelegt.				
Antwort	Das adCM-Werkzeug stellt dem Case Manager eine Funktion zur Verfügung, mit der er binäre und n-äre Beziehungen zwischen den Artefakten herstellen und beschriften kann.				
Antwort-metrik	Es gibt keine Artefakte, die sich nicht in eine binäre oder n-äre Beziehung setzen lassen. Die Beziehungen sind jederzeit für den Case Manager sichtbar.				
Szenario-ID	6	Anforderungs-ID	6, 7, 8	Quelle	Case Manager
Auslöser	Der Case Manager möchte ein Artefakt annotieren, das nicht vom eigenen Unternehmen bereitgestellt wird, beispielsweise einen Artikel, der auf der Webseite eines Ärzteverbandes veröffentlicht wurde.				
Umgebung	Der Case Manager hat mit dem adCM-Werkzeug ein extern bereitgestelltes Artefakt geöffnet. Das adCM-Werkzeug ist mit der Anwendung integriert, die für das Öffnen der Datei eingesetzt wird.				
Antwort	Das adCM-Werkzeug bietet in der Anwendung eine Benutzerschnittstelle zum Erstellen der Annotation an. Die Metadaten, aus denen sich die Annotation zusammensetzt, werden nicht in der Datei selbst, sondern separat in einem zentralen Datenspeicher abgelegt.				
Antwort-metrik	Das adCM-Werkzeug kann alle Arten von Artefakten annotieren, an die es angeschlossen wurde. Der Case Manager kann bei der Handhabung der Annotationen keinen Unterschied feststellen, ob die Anntoationen in dem Artefakt selbst oder zentral abgelegt werden.				

Eine genauere Betrachtung der Szenarien macht deutlich, dass sich manche Szenarien (1, 4 und 5) schwerpunktmäßig auf fachliche Entitäten und deren Darstellung auf der Benutzeroberfläche auswirken, wohingegen sich andere eher auf technische Konzepte wie Algorithmen oder Dienste beziehen, die unter der Oberfläche arbeiten (2, 3 und 6). Zum Beispiel wirkt sich das erste Szenario auf das Konzept der Agenda und dessen Darstellung aus. Das zweite Szenario hingegen definiert die Erwartungen, die an den Discovery-Algorithmus gestellt werden. Betrachtet man die

eher dienstbezogenen Szenarien im Detail, so lässt sich darüber hinaus feststellen, dass sie stets gewisse Anforderungen an die Integration des adCM-Werkzeugs stellen. Zum Beispiel muss für das Szenario 3 eine Anbindung an die unternehmensinternen Datenquellen bestehen und im Szenario 6 wird eine Integration des Werkzeugs in bestimmte Unternehmensanwendungen gefordert.

Insgesamt kann anhand der Szenarien herausgestellt werden, dass die zu gestaltende Architektur des adCM-Werkzeugs sowohl technische Schnittstellen als auch Benutzerschnittstellen gezielt berücksichtigen muss, um die volle Funktionalität der adCM-Konzepte bereitzustellen. Das adCM-Werkzeug ist also kein in sich abgeschlossenes System, das seine Funktionen ohne extern bereitgestellte technische Schnittstellen anbieten kann. Vielmehr dient es dem Zweck, die adCM-Konzepte Case Managern in einem bereits existierenden Arbeitsumfeld verfügbar zu machen. Es soll nicht bestehende Anwendungen ersetzen, die der Case Manager bereits nutzt. Das trifft insbesondere auf Office-Anwendungen zu, die zum Bearbeiten der Artefakte notwendig sind. Stattdessen soll es mithilfe der Agenda und der Workspaces ein planungsunterstützendes Portal zu unternehmensinternen und externen Datenquellen darstellen. Eine grundlegende architekturelevante Anforderung an die zu gestaltende übergeordnete Architektur des adCM-Werkzeugs ist daher, dass sie sich in eine bestehende IT-Unternehmensarchitektur integrieren lässt. Zu diesem Zweck müssen Schnittstellen zu den Unternehmensanwendungen einerseits und den Datenquellen andererseits existieren. Die Schnittstellen sollten darüber hinaus so beschaffen sein, dass die bestehende Architektur im Zuge der Integration nicht oder nur in möglichst geringem Umfang angepasst werden muss.

7.2.2 AUSWAHL UND ZUSAMMENSETZUNG DER ARCHITEKTURMUSTER

Um Architekturmuster zu identifizieren, die zu den oben beschriebenen Szenarien passen, wurden verschiedene Musterkataloge untersucht (Starke 2015; Vogel 2009; Buschmann 1996; Gamma 1995; Hohpe und Woolf 2004). Sie beschäftigen sich aus jeweils verschiedenen Perspektiven mit Architekturmustern. Beispielsweise werden in (Starke 2015) Muster auf einer Systemebene eingeführt, die zuvor in (Gamma 1995) als Muster auf Objektebene eingeführt wurden. Während der Recherche wurde gezielt nach Mustern gesucht, die sowohl die unterschiedlichen Schnittstellen zur Integration des Werkzeugs in die Unternehmen-IT als auch dessen Rolle als Portal für den Case Manager gezielt berücksichtigen. Als Ergebnis dieser Recherche wurden das Broker- und das Schichten-Architekturmuster ausgewählt. Nach welchem grundlegenden Prinzip beide Muster eine gegebene Systemstruktur gestalten und inwiefern sie eine konkrete Lösung für das vorliegende Problem darstellen, wird in den nachfolgenden Abschnitten beschrieben.

Gemäß dem Schichten-Architekturmuster werden die Komponenten in verschiedene Schichten mit unterschiedlichen Abstraktionsebenen aufgeteilt (Buschmann 1996, S. 31). Jede Schicht gruppiert dabei Komponenten, die Verantwortlichkeiten oder Funktionen auf einer ähnlichen Abstraktionsebene übernehmen. Die Einordnung in eine Schicht kann beispielsweise dadurch erfolgen, dass auf den oberen Schichten die Komponenten mit stark fachlich geprägten Verantwortlichkeiten gruppiert werden, während die Komponenten mit klar technischem Bezug auf den unteren Schichten zusammengefasst werden.

Übertragen auf die konkrete Problemstellung, nämlich der Integration der adCM-Lösungsbausteine in eine übergeordnete IT-Unternehmensarchitektur, bedeutet das, dass die Komponenten zur Anbindung der Unternehmens**daten** einerseits und der Unternehmens**anwendungen** andererseits in jeweils unterschiedliche Schichten eingeordnet werden. Im Ergebnis steht ein 4-Schichtenmodell (siehe Abbildung 54), auf dessen unterster Schicht sich alle Komponenten befinden, die die technische Bereitstellung von Datenquellen verantworten. Dazu zählen nicht nur unternehmensinterne Dokumentenserver oder Datenbanken, sondern auch extern bereitgestellte Webserver, auf die der Case Manager über das Internet zugreifen kann. Auf der obersten Ebene befinden sich die Anwendungen, die die fachlichen Geschäftsprozesse des Unternehmens unterstützen. Diese beiden Schichten stellen die Integrationspunkte mit dem adCM-Werkzeug dar. Denn dessen Lösungsbausteine befassen sich teils mit der Anbindung der Unternehmensdaten und teils mit der Integration von bestehenden Anwendungen, zum Beispiel durch die anwendungsübergreifende Annotationsfunktion. Sie bilden deshalb zwei zusätzliche Schichten, die zwischen den Daten und den Anwendungen liegen (siehe Abbildung 54, grau hinterlegt). Die einzelnen Schichten und deren Elemente werden in den folgenden Abschnitten im Detail erläutert.

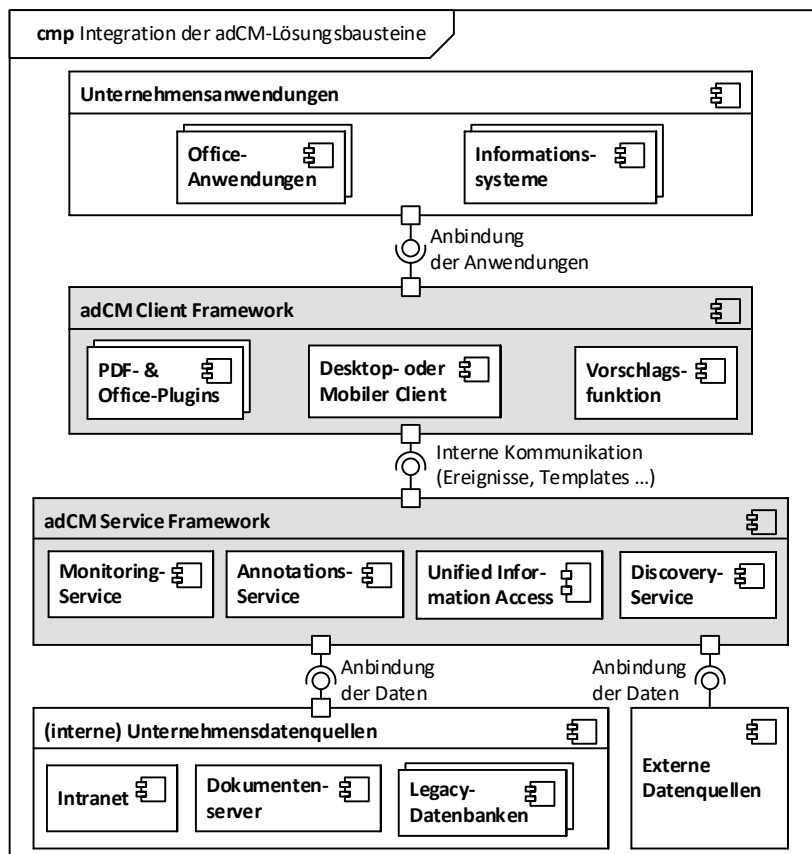


Abbildung 54: Integrationskonzept der adCM-Lösungsbausteine in eine Unternehmensarchitektur (UML-Komponentendiagramm)

Zu den **Datenquellen** auf der untersten Schicht zählen sowohl lokal (unternehmensintern) als auch außerhalb des Unternehmens gespeicherte Daten, die der Case Manager in Ausübung seiner Funktion benötigt oder produziert. Im Gegensatz zu strukturierten Prozessen, in denen Sachbearbeiter mit einer vorab bekannten und klar abgrenzbaren Menge an Daten arbeiten, ist eine solche Abgrenzung im Kontext schwach strukturierter Prozesse schwierig. Weil die benötigten Daten nicht immer bekannt sind, können die Elemente dieser Schicht nicht vollständig aufgezählt und konkret benannt werden. Es kann sich also beispielsweise um Intranet-Seiten, Wikis, Dokumente, Informationssysteme oder extern bereitgestellte Webseiten handeln. Die Elemente dieser Schicht sind nicht durch die adCM-Architektur, sondern durch die individuelle Unternehmensarchitektur vorgegeben (gekennzeichnet durch den weißen Hintergrund).

Die darüber liegende Schicht wird als **adCM Service Framework** bezeichnet. Dabei handelt es sich um eine Menge von austauschbaren und teils optionalen Services, die im Wesentlichen zwei Aufgaben erfüllen: erstens das Bereitstellen eines einheitlichen Portals zu allen Datenquellen (inkl. des gespeicherten Prozesswissens in Form von Templates) und zweitens das Persistieren von Daten aus dem Prozess (zum Beispiel die Ablage der Falldaten und das Protokollieren der Benutzerinteraktionen). Um diese Funktionen zu erfüllen, wurden die in Kapitel 5 und 6

beschriebenen Konzepte als einzelne Komponenten implementiert und in die Service Framework-Schicht integriert.

Die **adCM Client Framework**-Schicht besteht aus einem Desktop-Client, einer Vorschlagsfunktion (Recommender System), einer mobilen Anwendung und verschiedenen Plug-Ins. Der Client ist die zentrale Benutzerschnittstelle für alle im adCM enthaltenen Konzepte. Damit wird der Case Manager also unter anderem dazu befähigt, Fälle zu starten, Agenden zu pflegen und diese mit Artefakten zu versehen. Diese Konzepte werden prinzipiell auch von einer mobilen Anwendung unterstützt, allerdings zugeschnitten auf die im mobilen Kontext reduzierten Ressourcen und mit einem angepassten UI-Konzept. Die Plug-Ins dienen der Integration von adCM-Funktionen in Standard-Büroanwendungen, die zur Unterstützung bestimmter Kernaktivitäten wichtig sind. Das adCM Client Framework ist demnach aus Anwendersicht die Schnittstelle zu allen benötigten Datenquellen und adCM-Werkzeugen. Bezogen auf die Integration von adCM in ein Unternehmen ist diese Schicht die Schnittstelle zu der Anwendungslandschaft des Unternehmens.

Um dem Case Manager im Sinne eines zentralen Portals eine einheitliche Schnittstelle sowohl zu den Datenquellen als auch zu den Unternehmensanwendungen zu bieten, wird die Schichtenarchitektur mit dem **Broker-Architekturmuster** nach (Buschmann 1996, S. 99) kombiniert. Dieses Muster kommt zum Einsatz, wenn es sich um einen verteilten, heterogenen Systemkontext handelt, in dem die einzelnen Service- und Clientkomponenten zentral koordiniert werden müssen. Dabei erfüllt das adCM Framework in seiner Gesamtheit die Funktion des Brokers. Es koordiniert die Zugriffe des Case Managers auf die Artefakte und startet die dazugehörige Unternehmensanwendung. Wie diese Funktion im Detail erfüllt wird und wie auf diese Weise auch die einzelnen Kernaktivitäten des Case Managers unterstützt werden, ist Gegenstand der folgenden Kapitel.

7.3 DESIGN UND IMPLEMENTIERUNG DES MONITORING-SERVICES

Das Monitoring umfasst das Aufzeichnen von Ereignissen, die durch die Interaktion des Case Managers mit dem adCM-Werkzeug erzeugt werden. In diesem Abschnitt wird beschrieben, welche Design-Entscheidungen bei der Implementierung dieses Konzepts getroffen wurden und mit welchen Technologien das Design konkret umgesetzt wurde.

7.3.1 SZENARIEN FÜR ARCHITEKTURRELEVANTE ANFORDERUNGEN

Die Ereignisse aus den Benutzerinteraktionen sind nicht nur für die Protokollierung und die anschließende Analyse des Protokolls durch das Discovery relevant. Auch für das proaktive Vorschlagen von Templates während der Arbeit an einem Fall liefern die Ereignisse wichtige Informationen über den aktuellen Kontext des Falls. Es gibt also keinen zentralen Adressaten dieser Ereignisse. Darüber hinaus ist der adCM-Client nicht die einzige Komponente, die zu protokollierende Ereignisse verschickt. Ob ein Artefakt durchsucht oder annotiert wurde, können nur die Anwendungen feststellen, mit denen das Artefakt vom Case Manager bearbeitet wurde. Die Szenarien 7 und 8 verdeutlichen diese Zusammenhänge jeweils an einem Beispiel (siehe Tabelle 18).

Tabelle 18: Szenario zur Konkretisierung architekturentwicklungsrelevanter Anforderungen an den Monitoring-Service

Szenario-ID	7	Anforderungs-ID	2, 9, 12	Quelle	adCM-Client (Desktop/mobil)
Auslöser	Der adCM-Client registriert eine Benutzerinteraktion (zum Beispiel das Hinzufügen einer Agenda).				
Umgebung	Der adCM-Client wird vom Case Manager zur Steuerung eines Falls verwendet.				
Antwort	Der adCM-Client erzeugt ein Ereignis und sendet es an alle Komponenten, die es weiterverarbeiten sollen (zum Beispiel der Monitoring-Service und der Discovery-Service).				
Antwortmetrik	Alle Komponenten, die für die Weiterverarbeitung des Ereignisses verantwortlich sind, haben das Ereignis erhalten.				
Szenario-ID	8	Anforderungs-ID	9, 12	Quelle	PDF- oder Office-Plugin
Auslöser	Ein PDF- oder Office-Plugin registriert eine Benutzerinteraktion (zum Beispiel das Annotieren oder Durchsuchen eines Artefakts).				
Umgebung	Der Case Manager annotiert oder durchsucht ein Artefakt. Die dafür eingesetzte Anwendung ist mit einem adCM-Plugin versehen.				
Antwort	Das Plugin erzeugt ein Ereignis und sendet es an alle Komponenten, die es weiterverarbeiten sollen (zum Beispiel der Monitoring-Service und der Discovery-Service).				
Antwortmetrik	Alle Komponenten, die für die Weiterverarbeitung des Ereignisses verantwortlich sind, haben das Ereignis erhalten.				

Anhand der Szenarien lässt sich ablesen, dass die zu gestaltende Architektur eine Kommunikationsinfrastruktur zwischen den Komponenten bereitstellen muss, die sowohl mehrere Sender (Desktop-Client, mobiler Client, Plugins) als auch mehrere Empfänger (Monitoring-Service, Discovery-Service) verbinden kann und die den Versand unterschiedlicher Ereignisse unterstützt.

7.3.2 AUSWAHL UND ZUSAMMENSETZUNG DER ARCHITEKTURMUSTER

Für die oben beschriebenen Szenarien werden in der Literatur ereignisgesteuerte Architekturen vorgeschlagen (Starke 2015, S. 121; Dunkel et al. 2008, S. 119). Sie unterscheiden zwischen Produzenten und Konsumenten von Ereignissen, die über einen Kanal miteinander kommunizieren. Dabei sendet der Produzent das Ereignis nicht an einen konkreten Empfänger, sondern an den Kanal. Ein Konsument greift Ereignisse auf dem Kanal ab, die für seine Funktionalität relevant sind. Bezogen auf das Monitoring von Benutzerinteraktionen nehmen die Elemente des adCM Client Frameworks die Rolle der Ereignisproduzenten ein, während die Elemente des adCM Service Frameworks die Konsumenten sind (siehe Abbildung 55). Durch diese Architekturentscheidung wird das Prinzip der losen Kopplung (Starke 2015, S. 339) verfolgt.

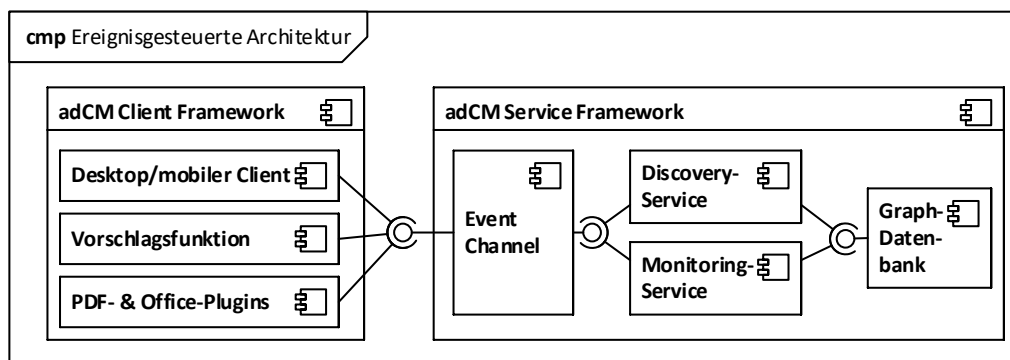


Abbildung 55: Kopplung von Client- und Service Framework mit einer ereignisgesteuerten Architektur (UML Komponentendiagramm)

Die Monitoring-Services greifen die Ereignisse auf dem Ereigniskanal ab, um die in Kapitel 5 beschriebene semantische Protokollierung zu realisieren. Die Discovery-Services setzen umgekehrt eine Nachricht auf dem Ereigniskanal ab, wenn die Erzeugung von Templates abgeschlossen ist. Diese Nachricht kann von der Vorschlagsfunktion abgefangen werden, um zum Beispiel den lokal zwischengespeicherten Satz an Templates zu aktualisieren.

Im folgenden Kapitel wird beschrieben, mit welchen Technologien diese ereignisgesteuerte Architektur umgesetzt wurde.

7.3.3 IMPLEMENTIERUNG DER EREIGNISGESTEUERTEN ARCHITEKTUR

Die oben beschriebene Art und Weise, wie die ereignisgesteuerte Architektur eingesetzt wird, orientiert sich an dem Publish/Subscriber-Interaktionsmuster, zu dem es eine Reihe unterschiedlicher Implementierungen (Eugster et al. 2003) und zahlreiche, teils technologiespezifische Standards gibt wie zum Beispiel die Java Message Service-Spezifikation. Weil bei der prototypischen Implementierung des adCM-Werkzeugs unterschiedliche

Technologien zum Einsatz kommen, musste ein technologieunabhängiger Standard verwendet werden. In der Literatur (Bloebaum und Johnsen 2015; Fernandes et al. 2013; Luzuriaga et al. 2015) werden vor allem zwei ISO/IEC-Standards für solche Architekturen diskutiert. Dabei handelt es sich um AMQP (Advanced Message Queuing Protocol), das im Standard ISO/IEC 19464:2014 veröffentlicht wurde sowie MQTT (Message Queueing Telemetry Transport), das als ISO/IEC 20922:2016 standardisiert ist. MQTT ist schwerpunktmäßig für den Einsatz in Netzwerken mit ressourcenschwachen Geräten ausgerichtet (Luzuriaga et al. 2015). AMQP weist im Gegensatz zu MQTT bei stabilen Netzwerken eine bessere Antwortzeit und bei instabilen Netzwerken eine bessere Zuverlässigkeit auf (Bloebaum und Johnsen 2015). Aus diesen Gründen fiel die Wahl auf AMQP.

Damit die Ereignisproduzenten und -konsumenten über das Protokoll Nachrichten austauschen können, muss bei der Implementierung dieser Architektur darauf geachtet werden, dass sie ein gemeinsames Thema (topic) verwenden. Das Thema stellt den Kanal dar, über den die Ereignisse transportiert werden. Die Ereignisproduzenten müssen die Ereignisse unter Angabe dieses Themas an den AMQP-Server senden und die Ereigniskonsumenten müssen dieses Thema am AMQP-Server abonnieren, um alle dazugehörigen Ereignisse zu erhalten.

7.3.4 *ADCM-SPEZIFISCHE ERWEITERUNG DES XES-STANDARDS*

Um die XES-konforme Protokollierung der Ereignisse zu implementieren, wurde eine Erweiterung des XES-Standards vorgenommen. Sie umfasst unter anderem die Konzepte des adCM-Metamodells. Auf welche Weise solche Erweiterungen spezifiziert werden können, ist ebenfalls Teil des XES-Standards (Verbeek et al. 2011). Demnach werden alle zusätzlichen Attribute in einer separaten XML-Datei definiert. Die XML-Datei (siehe Anhang 10.6) enthält im Wurzelement den Namen und das Präfix der Erweiterung, das später bei der Typisierung der zusätzlichen Attribute im Ereignisprotokoll verwendet wird (hier: „adcm“). Dem Wurzelement untergeordnet folgen zuerst die Attribute, die für Traces neu definiert werden. Weil das adCM-Werkzeug jeden Fall mit einer ID versieht, wird an dieser Stelle eine Erweiterung des Standards vorgenommen. Es folgen Erweiterungen der Ereignisattribute, die den Standard um die Eigenschaften aus dem in Kapitel 5 vorgestellten Ereignisdatenmodell ergänzen wie zum Beispiel den Typ des Ereignisses (gestartet, gestoppt, hinzugefügt) und den Typ der Ereignisquelle (Agenda, Fall, Artefakt).

Wie diese Erweiterung in den Metadaten der Protokolldatei eingebunden wird und wie die neuen Attribute eingesetzt werden, kann im angehängenen Protokoll eingesehen werden (siehe Anhang 10.2).

7.3.5 ARCHITEKTUR UND IMPLEMENTIERUNG DES MONITORING-SERVICE

Wie in Kapitel 5 beschrieben (vgl. auch Abbildung 20) werden die Ereignisdaten in einer serverseitigen Graphdatenbank gespeichert, um sie mit den adCM-Konzepten und gegebenenfalls auch domänenspezifischen Konkretisierungen dieser Konzepte zu verknüpfen. Für das Auslesen der Ereignisse aus dem Ereigniskanal der Message Oriented Middleware und der Persistierung dieser Daten in der Graphdatenbank ist der Monitoring-Service verantwortlich. Dessen Architektur ist in drei Schichten aufgeteilt: Kommunikation, Logik und Persistenz (siehe Abbildung 56).

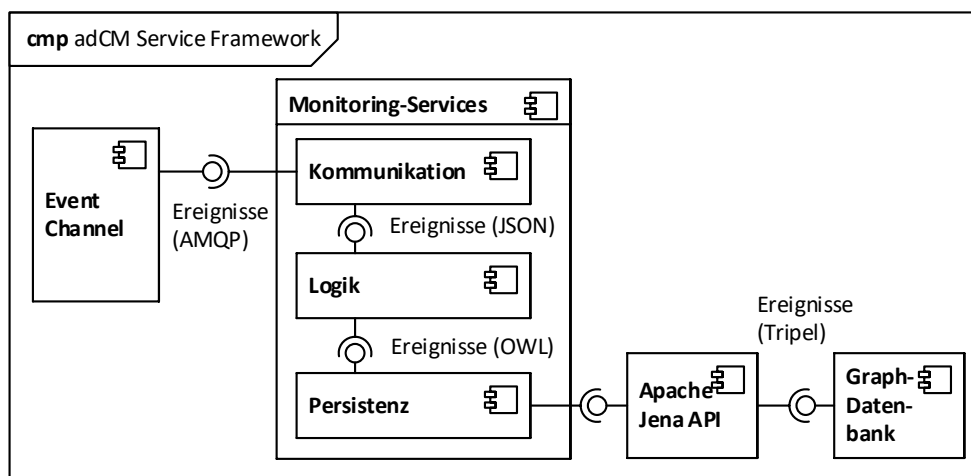


Abbildung 56: Architektur des Monitoring-Services (UML Komponentendiagramm)

Für das Auslesen der Ereignisdaten aus dem Event Channel ist in der Kommunikationsschicht die Komponente „MessageBusReader“ zuständig. Zu diesem Zweck implementiert sie den technologiespezifischen Nachrichtenaustausch über das AMQP-Protokoll. Sollte die Unternehmenslandschaft, in der das adCM-Werkzeug eingesetzt wird, eine andere Kommunikationstechnologie verwenden (zum Beispiel JMS), muss diese Implementierung ausgetauscht werden. Der MessageBusReader leitet die ausgelesenen Ereignisse, die vom Client als JSON-Objekte auf den Event Channel gelegt werden, weiter an die Logikschicht, ohne sie zu verarbeiten.

In der Logikschicht ist die Komponente „EventOWL“ dafür verantwortlich, die Ereignisdaten mit den Entitäten des adCM-Datenmodells zu verknüpfen und in OWL-Objekte (sog. „Individuals“) zu transformieren. Dazu werden die Attribute der JSON-Objekte in OWL-Properties umgewandelt. Je nach Ereignistyp wird außerdem eine Anwendungslogik angestoßen wie zum Beispiel das Übertragen eines Falls in die Menge der abgeschlossenen Fälle nach Eingang eines „case closed“-Ereignisses. Die OWL-Objekte werden zur Speicherung an die Persistenzschicht weitergeleitet.

Dort ist der „TripleStoreManager“ für die graphbasierte Persistierung der Ereignisdaten zuständig. In der prototypischen Implementierung wird als Graphdatenbank Apache Jena TDB eingesetzt. Für den Zugriff auf diese Datenbank bietet Apache Jena eine Programmierschnittstelle an, auf die der TripleStoreManager zugreift. Sollte in dem Unternehmenskontext eine andere Lösung zur Persistierung der semantisch verknüpften Ereignisdaten vorliegen, muss diese Komponente ausgetauscht werden.

Für weiterführende Details über die Implementierung der Monitoring-Komponente wird auf (Koop 2014) verwiesen.

7.4 DESIGN UND IMPLEMENTIERUNG DES DISCOVERY-SERVICE

In den nachfolgenden Abschnitten werden die Design-Entscheidungen zur Integration des Discovery-Algorithmus in das adCM Service Framework begründet. Dabei werden auch die Technologien benannt, die bei der Implementierung eingesetzt wurden.

7.4.1 SZENARIEN FÜR ARCHITEKTURRELEVANTE ANFORDERUNGEN

Wie der Discovery-Service in die Architektur eingeordnet werden sollte, hängt von der Art und Weise ab, wie der darin enthaltene Discovery-Algorithmus aufgerufen werden soll. Dabei kann zwischen zwei alternativen Szenarien unterschieden werden: Entweder wird der Algorithmus inkrementell nach jedem abgeschlossenen Fall ausgeführt (Szenario 9) oder zu festgelegten Zeiten gestartet (Szenario 10).

Tabelle 19: Szenario zur Konkretisierung architekturentwicklungsrelevanter Anforderungen an den Discovery-Service

Szenario-ID	9	Anforderungs-ID	14	Quelle	Case Manager
Auslöser	Ein Case Manager hat einen Fall abgeschlossen.				
Umgebung	Das adCM-Werkzeug wird im Produktivbetrieb eingesetzt, um Fälle zu planen. Der Discovery-Service ist als Ereigniskonsument ein Bestandteil der ereignisgesteuerten Architektur.				
Antwort	Der Client erzeugt und versendet das Ereignis „Fall abgeschlossen“. Die Discovery-Komponente liest das Ereignis aus und startet den Discovery-Algorithmus.				
Antwortmetrik	Der Algorithmus hat Templates auf Basis des aktuellen Stands aller abgeschlossenen Fälle erzeugt.				
Szenario-ID	10	Anforderungs-ID	14	Quelle	Externe Timer-Komponente
Auslöser	Ein Timer, der den Discovery-Algorithmus regelmäßig (zum Beispiel jedes Wochenende) auslöst, ist abgelaufen.				

Umgebung	Das adCM-Werkzeug wird im Produktivbetrieb eingesetzt, um Fälle zu planen. Der Discovery-Algorithmus kann über eine Schnittstelle des Discovery-Service angesteuert werden.
Antwort	Der Timer steuert den Discovery-Algorithmus direkt über eine explizit dafür vorgesehene Schnittstelle an.
Antwort-metrik	Der Algorithmus hat Templates auf Basis des aktuellen Stands aller abgeschlossenen Fälle erzeugt.

Wird das erste Szenario gewählt, muss der Discovery-Service das Ereignis „Fall abgeschlossen“ empfangen können. Für diesen Fall muss der Service also in der Rolle als Ereigniskonsument ein Bestandteil der ereignisgesteuerten Architektur sein. Soll der Algorithmus hingegen zu festgelegten Zeiten gestartet werden, muss der Discovery-Service nicht an diese Architektur angebunden sein. Stattdessen muss er eine Schnittstelle besitzen, über die eine externe Timer-Komponente den Algorithmus auslösen kann.

7.4.2 AUSWAHL UND ZUSAMMENSETZUNG DER ARCHITEKTURMUSTER

Eine inkrementelle Ausführung des Discovery-Algorithmus (Szenario 9) bietet im Gegensatz zu der zeitgesteuerten Ausführung (Szenario 10) zwei wesentliche Vorteile: Erstens kann auf diese Weise sichergestellt werden, dass die Templates immer den aktuellen Stand der Fallbasis widerspiegeln. Zweitens kann der Algorithmus so weiterentwickelt werden, dass er die Auswirkungen durch den hinzugefügten Fall auf die Mengenverhältnisse bei der Suche nach Templates eingrenzt und nur die notwendigen Veränderungen lokal verarbeitet. Dadurch ist eine Leistungssteigerung des Algorithmus zu erwarten, vor allem, wenn bereits sehr viele Fälle in der Fallbasis gesammelt wurden. Denn dann ist die Wahrscheinlichkeit umso niedriger, dass ein einzelner abgeschlossener Fall genau das Muster enthält, das nach den alten Mengenverhältnissen noch unter der Häufigkeitsschwelle war und erst durch den Abschluss dieses Falls ein neues Template erzeugt werden kann.

Ein im Produktivbetrieb eingesetztes adCM-Werkzeug sollte aufgrund der genannten Vorteile die inkrementelle Variante des Discovery-Algorithmus implementieren. In diesem Fall muss der Discovery-Service in die ereignisgesteuerte Architektur eingebunden sein. Für den Werkzeug-Prototyp, der sich auf die Darlegung der Machbarkeit der adCM-Konzepte konzentriert, stellen diese Optimierungen des Algorithmus aber keine architekturelevante Anforderung dar. Deshalb werden im weiteren Verlauf die Architekturentscheidungen und Implementierungsdetails beschrieben, die sich aus dem Szenario 10 ergeben. Dabei wird der Discovery-Algorithmus über einen einfachen Webservice angesteuert.

Die Architektur der Discovery-Services ist darüber hinaus, ebenso wie die Architektur der Monitoring-Services, in drei Schichten aufgeteilt, um die einzelnen Verantwortungsbereiche innerhalb der Komponente voneinander zu trennen (siehe Abbildung 57).

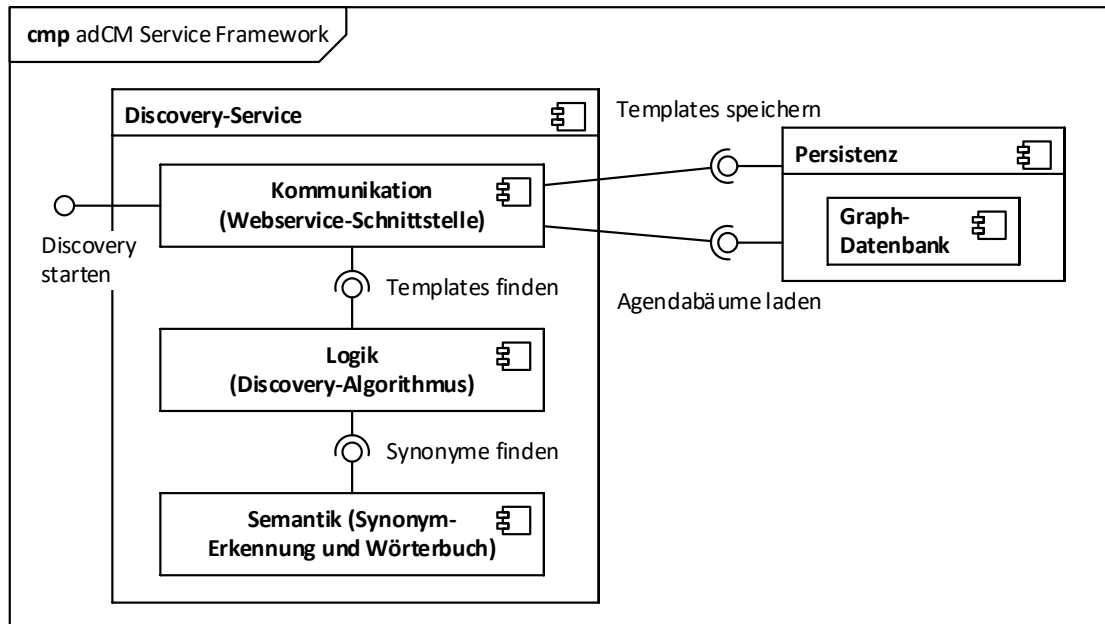


Abbildung 57: UML Kommunikationsdiagramm für den Aufruf des AgendaMiners (UML Komponentendiagramm)

Die Kommunikationsschicht stellt die Schnittstelle zu anderen Elementen der adCM-Architektur dar. Neben Schnittstellen zur Ansteuerung des Discovery-Algorithmus wird hier auch die bereits bestehende Persistenzschicht für den Zugriff auf die Wissensdatenbank angebunden. Diese Anbindung wird benötigt, um auf die Agendabäume zugreifen und die erzeugten Templates ablegen zu können.

Die Logikschicht enthält den in Kapitel 6 beschriebenen Discovery-Algorithmus (adCM Miner). Die für das semantische Clustering benötigte Synonymerkennung ist in einer separaten Schicht „Semantik“ ausgelagert. Welche Bibliotheken und Frameworks dabei zum Einsatz kommen, wird im nachfolgenden Implementierungskapitel beschrieben.

Weil es etablierte Standards gibt, die eine plattformunabhängige Definition und Verwendung von Webservices ermöglichen, erlaubt die Bereitstellung des Discovery-Algorithmus als Webservice eine sehr flexible Verwendung des Service. Aus Sicht des Discovery-Service spielt es keine Rolle, ob dessen Aufruf über eine Batchverarbeitung oder den Desktop-Client erfolgt und welche Betriebssysteme oder Programmiersprachen dabei jeweils eingesetzt werden.

Durch die Trennung der Kommunikationsschicht von der Logikschicht kann die Implementierung der Webservice-Schnittstelle beliebig ausgetauscht werden. Beispielsweise kann eine RPC-Implementierung durch SOAP oder REST ausgetauscht werden (Xinyang Feng et al. 2009). Die Trennung der Logikschicht von der Semantiksicht vereinfacht den Austausch der verwendeten Algorithmen und Datenbanken zur Identifizierung von Synonymen in Agendaeinträgen.

7.4.3 IMPLEMENTIERUNG DER DISCOVERY-SERVICE

Die Bereitstellung der Discovery-Services wurde in der prototypischen Implementierung mithilfe einer REST-Schnittstelle implementiert (Kalinowski 2013). Diese Technologieentscheidung fußt auf der Feststellung, dass die Kommunikation mit den Discovery-Services zustandslos ist, einem wesentlichen Merkmal einer REST-Schnittstelle (Xinyang Feng et al. 2009). Zustandslos bedeutet in diesem Zusammenhang, dass

- a) der Aufruf des Discovery-Algorithmus nicht von anderen Aufrufen der Schnittstelle abhängig ist und
- b) über die Dauer der Verarbeitung hinaus keine Informationen über einen Aufruf gespeichert werden.

Im Folgenden wird der Ablauf beschrieben, der nach Aufruf der REST-Schnittstelle durch die Komponenten dieser Schichten führt. Er wird in einem UML-Kommunikationsdiagramm dargestellt (siehe Abbildung 58). Aus Gründen der Übersichtlichkeit sind die Antwortnachrichten ausgeblendet worden.

Die Interaktion beginnt mit dem Aufruf des Discovery-Algorithmus (AgendaMiner) durch die entsprechende Methode in der REST-Schnittstelle (1). Um die Templates zu gewinnen, lädt der Algorithmus zu Beginn den Agenda-Wald aus der Graph-Datenbank (1.1). Hierzu greift er über die Komponente „OntologyHandler“ auf die REST-Schnittstelle der Graphdatenbank zu (1.1.1). Im nächsten Schritt gibt der AgendaMiner das Clustering der Agendaeinträge bei der dafür verantwortlichen Komponente in Auftrag (1.2). Für das Clustering werden synonyme Agendaeinträge zusammengefasst. Hierzu werden zu jedem Agendaeintrag über das Java Wordnet Interface (JWI) passende Synonyme geladen (1.2.1, 1.2.1.1). Nach dem Clustering wird der eigentliche Discovery-Algorithmus mit den über die REST-Schnittstelle angegebenen Parametern ausgeführt (1.3). Die so gewonnenen Templates werden über die REST-Schnittstelle „OntologyHandler“ in die Graphdatenbank gespeichert (1.4, 1.4.1).

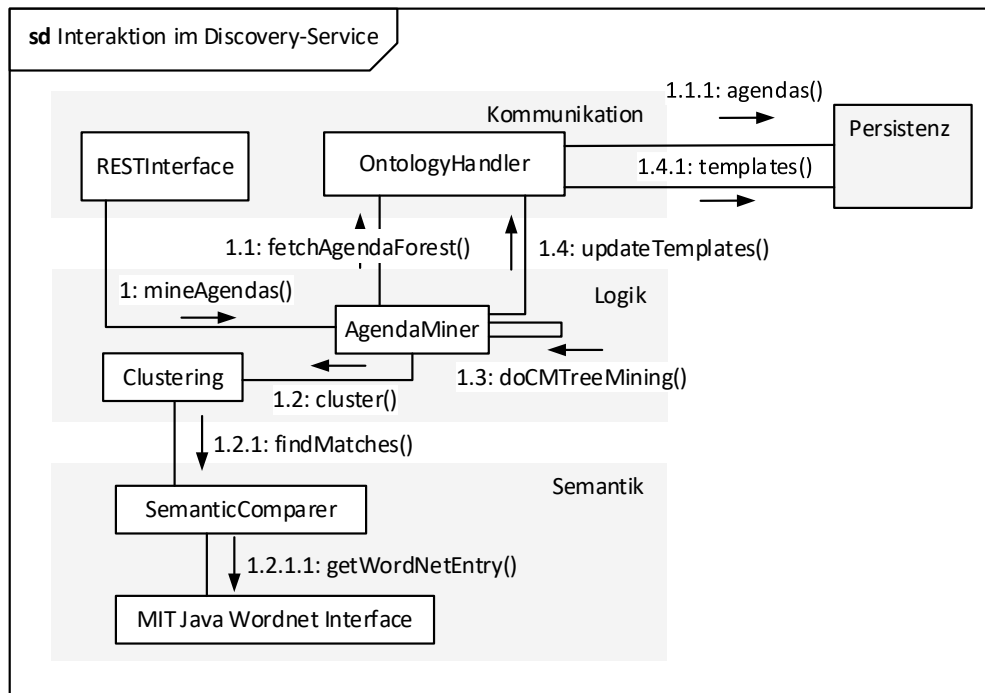


Abbildung 58: Darstellung der Interaktion mithilfe eines UML-Kommunikationsdiagramms

Jedes Cluster repräsentiert eine Reihe von Synonymen. Um einen passenden Cluster für einen Agendaeintrag zu finden, wird ein Leader-Algorithmus (Hartigan 1975) verwendet. Bei der Auswahl der WordNet-API wurde JWJ anderen Bibliotheken wie WS4J (WordNet Similarity for Java) vorgezogen, weil deren Ähnlichkeitsmetriken nicht in der Lage waren, Antonyme – also gegensätzliche Begriffe – zu trennen (Kalinowski 2013).

7.5 DESIGN UND IMPLEMENTIERUNG DER VORSCHLAGSFUNKTION

Im Process Mining-Forschungsfeld ist das Vorschlagen von erlaubten Folgeaktivitäten basierend auf einem Prozessmodell dem „Operational Support“ zuzuordnen (van der Aalst 2011). Im Kontext strukturierter Geschäftsprozesse werden WfMS für diesen Zweck eingesetzt. Mithilfe der vorliegenden Prozessmodelle, in denen alle erlaubten Pfade abgebildet sind, können solche Aktivitäten vorgeschlagen werden unter der Voraussetzung, dass der Zustand des Prozesses (im Sinne der aktuellen Position im Modell) stets bekannt ist.

Im Kontext schwach strukturierter Prozesse, die von adCM unterstützt werden, wird der Case Manager hingegen von einer sich dem Prozessverlauf anpassenden Agenda und einer Menge unzusammenhängender Templates unterstützt, anstelle eines imperativen Prozessmodells. Deshalb wurde die Vorschlagsfunktion so konzipiert, dass die Relevanz der verfügbaren Templates kontinuierlich neu bewertet wird unter ständiger Beobachtung der Veränderungen an der Agenda. Sie ist Bestandteil des adCM Client-Frameworks (siehe Abbildung 55). Im weiteren

Verlauf wird im Detail und auf Grundlage eines Szenarios beleuchtet, welche Entwurfsentscheidungen dabei getroffen wurden.

7.5.1 SZENARIEN FÜR ARCHITEKTURRELEVANTE ANFORDERUNGEN

Das Vorschlagen mutmaßlich nützlicher Aktivitäten während der Prozessausführung unterliegt der Herausforderung, für einen Vorschlag sowohl den richtigen Zeitpunkt als auch die richtige Zusammenstellung an Templates zu finden. Beide Aspekte werden durch die nachfolgenden Szenarien beschrieben.

Szenario-ID	11	Anforderungs-ID	2	Quelle	Case Manager
Auslöser	Ein Case Manager erweitert eine Agenda um Einträge, die in der Vergangenheit häufig verwendet wurden.				
Umgebung	Die aktuell bearbeitete Agenda stimmt mit mindestens einem Template in vielen Punkten überein.				
Antwort	Die Vorschlagsfunktion bietet dem Case Manager passende Templates an.				
Antwort-metrik	Der Zeitpunkt des Vorschlags ist so gewählt, dass der Case Manager nicht während einer kognitiv anspruchsvollen Aktivität unterbrochen wird. Zudem ist nachvollziehbar, dass die vorgeschlagenen Templates tatsächlich zum aktuellen Fall passen.				
Szenario-ID	12	Anforderungs-ID	2	Quelle	Case Manager
Auslöser	Ein Case Manager erweitert eine Agenda um Einträge, die in der Vergangenheit häufig verwendet wurden.				
Umgebung	Die aktuell bearbeitete Agenda stimmt mit einer Vielzahl von Templates in vielen Punkten überein.				
Antwort	Die Vorschlagsfunktion bietet dem Case Manager passende Templates an.				
Antwort-metrik	Die Menge an vorgeschlagenen Templates ist so gewählt, dass dem Case Manager nicht versehentlich das am Besten passende Template vorenthalten wird, der Case Manager aber die Anzahl noch überblicken kann.				

Das Szenario 11 macht deutlich, dass bezogen auf die Wahl des Zeitpunkts eines Vorschlags eine Balance gefunden werden muss zwischen der Effektivität und der Unaufdringlichkeit der Methode: Einerseits soll der Case Manager während der Ausführung kognitiv anspruchsvoller Aktivitäten nach Möglichkeit nicht unterbrochen werden, andererseits sinkt die Effektivität der Unterstützungsfunktion, wenn ein Template mit passenden Aktivitäten zu spät vorgeschlagen wird. Gemäß Szenario 12 muss bezogen auf die richtige Zusammenstellung ebenfalls eine Balance gefunden werden: Je mehr Templates vorgeschlagen werden, desto schwieriger ist die Identifikation eines relevanten Template durch den Case Manager. Durch die Auswahl geht wertvolle Zeit verloren, die der Case Manager andernfalls in die Fallsteuerung investiert hätte.

Andererseits – je weniger Templates vorgeschlagen werden, desto höher wird die Wahrscheinlichkeit, dass das tatsächlich passende Template dem Case Manager vorenthalten wird.

7.5.2 AUSWAHL UND ZUSAMMENSETZUNG DER ARCHITEKTURMUSTER

Die Szenarien zeigen, dass das Verhalten der Vorschlagsfunktion stark vom Zustand des Systems abhängig ist. Es muss ein kontinuierlicher Abgleich zwischen der aktuellen Agenda und der Menge passender Templates stattfinden. Erst wenn bestimmte Bedingungen erfüllt sind, sollen Templates vorgeschlagen werden. Nach Durchsicht der Musterkataloge wurde ein Muster aus der objektorientierten Programmierung identifiziert, dessen Problemkontext zu den beschriebenen Szenarios passt: der Zustand (Gamma 1995, S. 305). Gemäß einer Interpretation dieses Musters auf Systemebene (Starke 2015, S. 136) wird das Muster für Komponenten eingesetzt, die je nach Zustand ein spezielles Verhalten zeigen sollen. In Anlehnung daran wurde der Recommender-Algorithmus mit einem Zustandsautomaten versehen, der in Abhängigkeit vom Kontext der Agenda zwischen drei Stufen der Unterstützung unterscheidet (Benner-Wickner et al. 2015a). Mit jeder Stufe steigt gewissermaßen die Gewissheit des Algorithmus, dass das Vorschlagen von Templates in dem aktuellen Kontext sinnvoll ist. Wie die Zustandsübergänge zwischen den einzelnen Stufen definiert sind und welchen Grad an Unterstützung die Recommender-Funktion in dieser Stufe bietet, wird im weiteren Verlauf dargelegt.

Stufe 0: Wenn der Case Manager einen neuen Fall gestartet hat, ist die Agenda noch leer oder enthält nur wenige Agendaeinträge. Dann befindet sich der Algorithmus im Initialzustand. In diesem Zustand ist die Agenda so klein, sodass noch nicht genügend Kontextinformationen vorliegen um bewerten zu können, welche Templates zu dem Fall passen. Konsequenterweise ist der Algorithmus in diesem Zustand passiv und beobachtet lediglich die fortschreitende Herausbildung der Agenda. Der Übergang zu Stufe 1 erfolgt, sobald die Größe der Agenda, ausgedrückt durch die Anzahl der Agendaeinträge, einen vorgegebenen Schwellwert m erreicht. In der prototypischen Implementierung, die auch für die Evaluation des Algorithmus in Kapitel 6.4 zum Einsatz kam, wurde der Wert $m=3$ als Schwelle gewählt. Bei der Evaluation hat sich gezeigt, dass dieser Wert bereits ausreicht, um die Menge an potentiell relevanten Templates einzuschränken. In der Praxis muss dieser Wert anhand der User Experience angepasst werden.

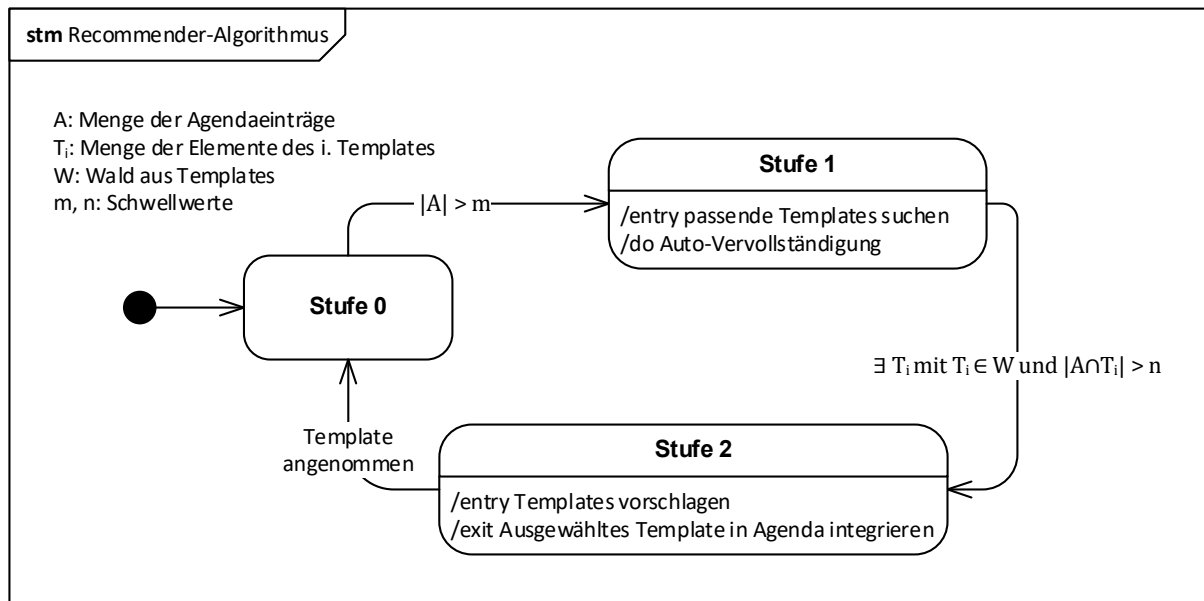


Abbildung 59: Vereinfachte Darstellung des Lebenszyklus des dreistufigen Recommender-Algorithmus (UML Zustandsdiagramm)

Stufe 1: Die zur Verfügung stehenden Kontextinformationen sind in dieser Stufe zwar ausreichend, um die Menge aller Templates deutlich zu reduzieren. Es sind aber trotzdem noch zu viele, um sie alle dem Case Manager vorzuschlagen (am Beispiel der Evaluation lag die Zahl stets im unteren zweistelligen Bereich). Stattdessen wird die Auswahl an Templates genutzt, um eine Autovervollständigungsfunktion beim Hinzufügen von Agendaeinträgen mit Schlagwörtern zu beliefern. Die Schlagwörter entsprechen den Template-Elementen. Auf diese Weise werden die Inhalte der Templates bereits genutzt mit dem Ziel, den Case Manager zwar bei der Planung seiner Aktivitäten zu unterstützen, jedoch ohne ihn durch einen Vorschlagsdialog in seiner Arbeit zu unterbrechen. Jedes Mal, wenn der Case Manager die Vorschläge in der Autovervollständigung annimmt, bestätigt er gewissermaßen die getroffene Auswahl der Templates. Den daraus ableitbaren Zugewinn an Gewissheit über die Relevanz der Templates macht der Algorithmus messbar, indem er einen kontinuierlichen Abgleich zwischen den Agendaeinträgen und den Template-Elementen durchführt. Sobald genügend Übereinstimmungen (mehr als n) zwischen der aktuellen Agenda und mindestens einem Template besteht, wechselt der Algorithmus in den nächsten Zustand. Auch hierfür ist die Vorgabe eines Schwellwertes für die Zahl der Übereinstimmungen notwendig. Bei der Evaluation mit der prototypischen Implementierung wurde der Schwellwert auf $n=3$ übereinstimmende Elemente festgelegt. Diese Zahl kann erhöht werden, um die Präzision des Algorithmus zu verbessern. Im Gegenzug sinkt aber die Fitness, weil große Teile der passenden Templates bereits in der Agenda enthalten sind, ohne dass der Recommender-Algorithmus die Gelegenheit bekam, sie vorzuschlagen.

Stufe 2: Das Überschreiten des Schwellwerts interpretiert der Algorithmus als Signal, dass nun eine hinreichende Relevanz der betroffenen Templates gegeben ist. Der Kontext des Falls (im Sinne der aktuellen Ausprägung der Agenda) hat sich zu diesem Zeitpunkt mindestens einem Template so stark angenähert, dass der Algorithmus zu einer anderen Strategie wechselt. Ab dann werden die Templates nämlich nicht nur für die Belieferung der Autovervollständigung eingesetzt, sondern mit einem separaten Dialog gezielt dem Case Manager angeboten. Es ist dem Case Manager überlassen,

- a) welches der vorgeschlagenen Templates er übernehmen möchte,
- b) an welche Stelle in der Agenda das Template integriert werden soll und
- c) ob das Template als Ganzes oder nur ausgewählte Elemente importiert werden sollen.

Mit dem oben beschriebenen dreistufigen Algorithmus wird die Intensität der Unterstützung angepasst an die Gewissheit über die Relevanz der zuvor gewonnenen Templates. Dadurch löst er das Problem, einen passenden Zeitpunkt für eine Unterbrechung der Arbeit des Case Managers durch das Vorschlagen von Templates zu finden.

Weil der Case Manager selbst entscheiden kann, welches Templates und welche Elemente er daraus übernimmt, wird ihm die nötige Flexibilität bei der Ausgestaltung der Agenda geboten. In diesem Sinne orientiert sich der Vorschlagsalgorithmus an dem grundlegenden Flexibilitätsprinzip des adCM-Konzepts.

7.6 DESIGN UND IMPLEMENTIERUNG VON FUNKTIONEN ZUR UNTERSTÜTZUNG WEITERER KERNAKTIVITÄTEN

Die oben vorgestellten Komponenten der adCM-Architektur beziehen sich hauptsächlich auf die Unterstützung des Case Managers beim Planen der nächsten Arbeitsschritte. In diesem Unterkapitel werden Architekturentscheidungen beschrieben, die anknüpfend daran auch Komponenten für die Unterstützung weiterer Kernaktivitäten umfassen. Ihnen liegen die Szenarien 3-6 aus Tabelle 17 zugrunde.

7.6.1 UNTERSTÜTZUNG BEIM RECHERCHIEREN

Gemäß Kapitel 2 besteht eine der Kernaktivitäten eines Wissensarbeiters darin, umfangreiche Datenbestände nach wertvollen Informationen zu durchsuchen (recherchieren). In diesem Abschnitt werden die Architekturentscheidungen für die Unterstützung dieser Aktivität beschrieben.

7.6.1.1 Zentraler Service für den Datenzugriff

Weil in schwach strukturierten Prozessen kaum Datenquellen a priori ausgeschlossen werden können, muss der Case Manager entsprechend viele unterschiedliche Portale beziehungsweise Zugänge verwenden können. In diesem Zusammenhang wurde die Anforderung 3 an das adCM-Konzept gestellt und im Szenario 3 konkretisiert. Es besagt, dass das System dem Case Manager den Zugriff auf diese Vielzahl an Datenquellen vereinfachen soll. Im adCM Service Framework wird diese Funktion durch eine Komponente realisiert, die auf Technologien aus dem Themengebiet des Unified Information Access (Schatten et al. 2003) zurückgreift (Benner-Wickner et al. 2014b).

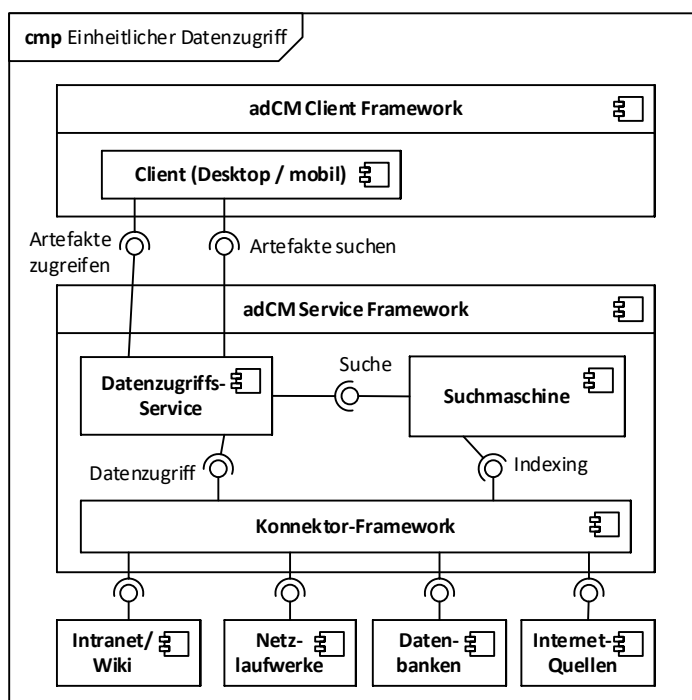


Abbildung 60: Architektursicht auf den einheitlichen Datenzugriff (UML Komponentendiagramm)

Zur Vereinheitlichung des Zugriffs auf die vielen unterschiedlichen Datenquellen bietet die Komponente einen zentralen Service, der von der jeweiligen Benutzerschnittstelle im adCM Client Framework verwendet wird (siehe Abbildung 60). Dieser Zugriff erfolgt entweder über den Workspace, zum Beispiel durch das Auswählen eines als Verknüpfung abgelegten Artefakts, oder über eine separate Suchmaske. Der Service ist unmittelbar verbunden mit einer Volltextsuchmaschine und einem Framework aus speziellen Konnektoren für die jeweiligen Datenquellen. In der prototypischen Implementierung kommt als Suchmaschine die quelloffene

Software Solr⁷ zum Einsatz. Ihr Kern ist die freie Java-Bibliothek Lucene⁸, mit der sie zusammen als Apache-Projekt weiterentwickelt wird. Lucene stellt die Algorithmen sowohl für die Indizierung als auch für die Suche bereit. Solr ist nach (solidIT 2015) die bekannteste Suchmaschinensoftware und findet in vielen wissenschaftlichen Projekten Anwendung (Hardesty und Greene 2013; Chen et al. 2012; Lacic et al. 2014; Artini et al. 2014).

Um die Suchmaschine mit allen Datenquellen eines Unternehmens zu verbinden, muss je Quelle ein eigener Konnektor entwickelt und integriert werden. Um diesen Aufwand zu vermeiden, schlägt die Architektur die Verwendung eines Konnektor-Frameworks vor, das Konnektoren für eine große Zahl an möglichen Datenquellen bereits integriert. In der prototypischen Implementierung wird ManifoldCF⁹ eingesetzt, welches – wie auch Solr und Lucene – von einem Apache-Projekt gepflegt wird. Das Framework bietet neben den vorgefertigten Konnektoren eine API an, mit der ein Unternehmen zusätzliche Konnektoren selbst erstellen und hinzufügen kann. Um bei der Suche die Zugriffsrechte des Case Managers bei den angebundenen Datenquellen zu berücksichtigen, bietet das Framework eine Anbindung an einen Active Directory-Service an.

7.6.1.2 Anbindung von Datenbanken mithilfe virtueller Dokumente

Die meisten Datenquellen teilen die Eigenschaft, dass ihre Elemente mithilfe eines Pfades (URL) direkt referenziert werden können. In solchen Fällen verweisen die Artefakte auf dem Workspace oder die Suchergebnisse der Volltextsuche unmittelbar auf die entsprechenden Dateien oder Webseiten. Allerdings gibt es auch Quellen wie zum Beispiel Datenbanken von Informationssystemen, deren Inhalte sich nicht auf eine URL abbilden lassen. Solche Systeme stellen die Informationen zudem erst dynamisch aus verschiedenen Datenbanktabellen zusammen. Um dieses Problem zu lösen, wird das Konzept der virtuellen Dokumente aufgegriffen (Zhang et al. 2010). Dabei handelt es sich um Schablonen, die dynamische Inhalte aus Datenbank in Dateiform repräsentieren können. Genau genommen enthalten die Schablonen eine beliebige komplexe Datenbankabfrage, die zum Zeitpunkt des Zugriffs auf das virtuelle Dokument ausgelöst wird und deren Ergebnis in einer HTML-Datei ausgegeben wird. Für den Anwender stellen sich virtuelle Dokumente in den Suchergebnissen oder auf dem Workspace wie übliche Dateien beziehungsweise Artefakte dar.

⁷ <http://lucene.apache.org/solr/>

⁸ <https://lucene.apache.org/core/>

⁹ <https://manifoldcf.apache.org/>

7.6.2 *UNTERSTÜTZUNG BEIM ARCHIVIEREN*

Hat der Case Manager alle benötigten Daten im Zugriff, so besteht gemäß dem Szenario 4 die nächste Herausforderung darin, eine integrierte Ablage für diese heterogenen Daten bereitzustellen. In diesem Abschnitt werden Architekturentscheidungen motiviert, die eine solche Unterstützungsfunktion ermöglichen.

7.6.2.1 **Zentrale Datenablage**

Alle einem Fall hinzugefügten Daten werden in adCM als Artefakte bezeichnet. Diese Artefakte liegen gemäß dem Metamodell in einem Workspace, der einem Agendaeintrag zugeordnet ist. Ein Fall enthält demnach eine Reihe von hierarchisch angeordneten Workspaces, deren Inhalte bei einer Persistierung des Falls berücksichtigt werden müssen. Im Kapitel 6 wurde bereits bei der Vorstellung des Datenmodells kurz auf die Speicherung von Falldaten eingegangen. Dabei fiel die Wahl auf einen graphbasierten Ansatz, der zugunsten einer besseren Flexibilität dem relationalen Ansatz vorgezogen wurde. Zwar werden in diesem Graph auch die Artefakte berücksichtigt, es ist aber noch nicht darauf eingegangen worden, ob die Artefakte dort nur im Sinne einer Verknüpfung oder inklusive der enthaltenen Daten gespeichert werden. Diese Entscheidung ist architekturelevant und wird hier deshalb kurz beschrieben.

Artefakte können aufgrund ihrer Definition im Meta-Modell sehr unterschiedlich in ihrer Größe und ihrer Strukturiertheit sein. Graphbasierte Datenbanken sind aber auf die Speicherung von semantisch verknüpften Informationen ausgelegt, nicht von großen binären Daten. Deshalb unterscheidet die adCM-Architektur bei der Datenablage zwischen dateibasierten und nicht-dateibasierten Artefakten. Dateibasierte Artefakte wie zum Beispiel Office-Dokumente, werden demnach in einem Dateisystem gespeichert, auf dessen Pfad im semantischen Datenmodell verwiesen wird. Je Fall wird ein eigener Ordner angelegt, sodass die Zugriffsrechte der einzelnen Case Manager berücksichtigt werden können. Nicht-dateibasierte Artefakte, beispielsweise Annotationen oder Links, werden hingegen ausschließlich und vollständig im Datenmodell gespeichert.

Gemäß der Anforderung 4 ist bei den Architekturentscheidungen zur Gestaltung der Datenablage auf die Verfügbarkeit und Performanz zu achten. Deshalb können die Artefakte lokal auf dem jeweiligen Client abgelegt werden. Dadurch ist gewährleistet, dass die Arbeit des Case Managers nicht von der Verfügbarkeit einer kontinuierlichen Datenverbindung oder einer zuverlässigen Synchronisation der Daten abhängig ist. Zudem ist ein lokaler Zugriff auf die Artefakte schneller als ein Zugriff über ein entferntes System.

In der prototypischen Implementierung der adCM-Architektur wird es dem Benutzer überlassen, welche Artefakte lokal abgelegt werden. Diese Entscheidung kann zu zwei Zeitpunkten getroffen werden:

1. beim Import des Artefakts in den Workspace: In diesem Fall wird die Datei nicht als Verknüpfung, sondern als lokal gespeicherte Datei im Workspace erstellt.
2. nachträglich während der Bearbeitung des Falls: Verknüpfungen auf dem Workspace verfügen über einen speziellen Eintrag im Kontextmenü, mit dessen Hilfe eine Funktion aufgerufen werden kann, die den Pfad zum Artefakt auflöst und die Inhalte herunterlädt.

7.6.2.2 Synchronisation der Ablage

Weil das adCM Client Framework den Zugriff über mehrere Clients unterstützt (zum Beispiel über eine mobile Anwendung), müssen die lokal gespeicherten Artefakte zwischen den Clients synchronisiert werden. Eine verbreitete Technologie zur Synchronisierung von Dateien sind sogenannte Cloud Storage-Technologien (Drago et al. 2012). Sie bestehen aus einem extern bereitgestellten Datenspeicher und einer Anwendung zur Synchronisierung der lokalen Daten mit dem externen Datenspeicher. Ursprünglich entwickelt für den persönlichen Gebrauch, werden diese Dienste nunmehr gezielt auch für den Einsatz in einem Unternehmen angeboten. Dabei ist ein vielfältiges Angebot verschiedener Dienstleister wie Microsoft und Google entstanden. Alternativ dazu können Unternehmen eine eigene Cloud Storage-Infrastruktur bereitstellen, zum Beispiel mit freier Software wie OwnCloud¹⁰.

Aufgrund dieses vielfältigen Angebots ist die Einführung eines neuen Synchronisierungsmechanismus nicht notwendig. Stattdessen soll das adCM-Werkzeug die vom Unternehmen eingesetzte Cloud Storage-Technologie mitbenutzen. Zu diesem Zweck sieht die Architektur des adCM-Werkzeugs ein Adapter-Muster vor (Gamma 1995). Es bietet eine Schnittstelle mit Grundfunktionen von Synchronisierungsmechanismen, die von der konkreten Implementierung abstrahieren. Um einen Machbarkeitsnachweis zu dieser Architekturentscheidung zu liefern, wurden beispielhaft zwei Adapter für je eine Cloud Hosting-Lösung implementiert und in den Prototyp des adCM-Werkzeugs integriert. Ein Adapter verbindet das Werkzeug mit einer von Dropbox bereitgestellten Datenablage, der zweite Adapter mit einer Owncloud-Ablage. Die Abbildung 61 zeigt mithilfe eines UML-Klassendiagramms, wie die Adapter implementiert und in das Werkzeug integriert wurden. Ganz oben ist die Klasse „SynchronizationMethod_Remote“ zu sehen, die in der prototypischen Implementierung des

¹⁰ <https://owncloud.org/>

adCM-Werkzeugs die Synchronisation der Artefakte auslöst. Sie verwendet zu diesem Zweck die Schnittstelle „IAdapter“, die je nach Unternehmenskontext von unterschiedlichen Adaptern implementiert wird (hier: Owncloud und Dropbox). Weil es bereits die Programmibliothek „DropNetClient“ für den Zugriff auf den Dropbox-Speicher gibt, muss der DropboxAdapter lediglich die in der Adapter-Schnittstelle definierten Methodenaufrufe an die Bibliothek delegieren.

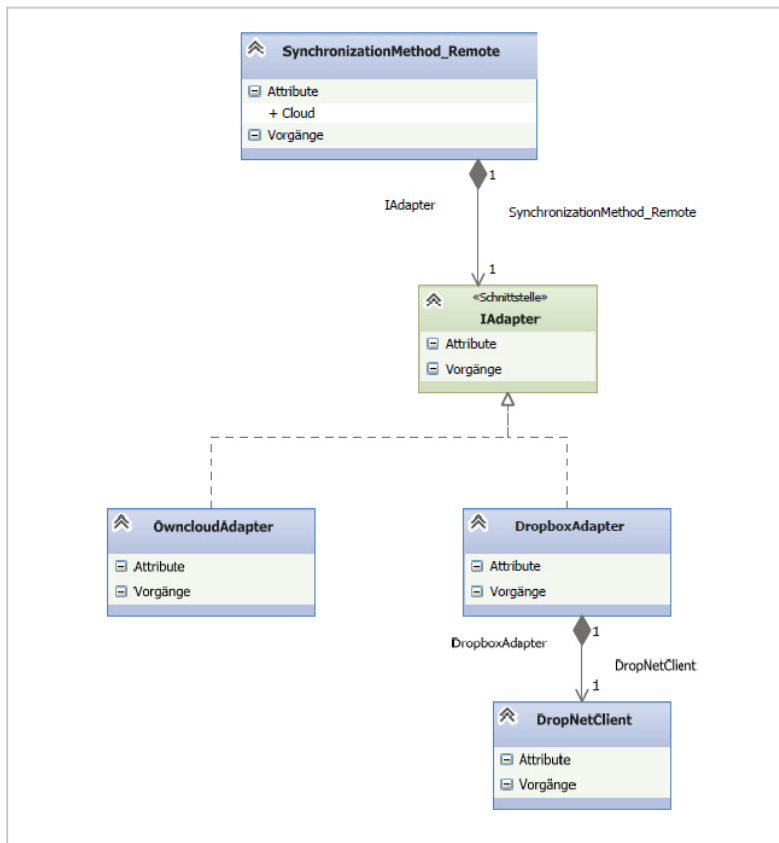


Abbildung 61: Adapter-Architektur zur Anbindung verschiedener Synchronisationsmechanismen

7.6.3 UNTERSTÜTZUNG BEIM STRUKTURIEREN

Gemäß Anforderung 5 und dem dazugehörigen Szenario aus Tabelle 17 soll der Wissensarbeiter auch dahingehend unterstützt werden, dass er Artefakte miteinander in Beziehung setzen kann. In diesem Abschnitt wird beschrieben, mit welchen Design-Entscheidungen diese Form der Strukturierung von Artefakten unterstützt werden.

7.6.3.1 Visualisieren von Beziehungen zwischen Artefakten

In der prototypischen Implementierung wird unterschieden zwischen binären und n-ären Beziehungen. Sie können eine Semantik haben, die vom Case Manager per Freitexteingabe vergeben wird. Zum Beispiel kann ein Heilverlaufsplan mit einem Gutachten in Beziehung

gebracht werden, wobei das Gutachten den Heilverlaufsplan um Detailinformationen zu einer therapeutischen Maßnahme verfeinert. Diese Semantik kann der Case Manager über eine binäre Beziehung, visualisiert durch eine Verbindung und über das Stichwort „verfeinern“, ausdrücken. Handelt es sich um mehrere Artefakte, die zu einer gemeinsamen Semantik gehören, kann dieser Zusammenhang mit einer Gruppierung hergestellt werden. Sie wird nicht mit einer Verbindung, sondern mit einem Kreis visualisiert, der die Artefakte umschließt.



Abbildung 62: Binäre (links) und n-äre (rechts) Beziehungen zwischen Artefakten auf dem Workspace

7.6.3.2 Design der Desktop-Benutzeroberfläche zum Strukturieren der Artefakte und Agendaeinträge

Um die zentrale Datenablage mithilfe der Workspaces in geeigneter Weise zu visualisieren, wurden mehrere unterschiedliche Ansätze gegenübergestellt und teils prototypisch implementiert. Im weiteren Verlauf dieses Abschnitts werden diese Ansätze gegenübergestellt.

Der erste Ansatz besteht darin, die Agenda, den jeweils ausgewählten Workspace und einen Artefakt-Browser zugleich auf der Oberfläche anzuzeigen (Benner et al. 2012). Je nachdem, welcher Agendaeintrag aktuell durch den Case Manager ausgewählt ist, wird der dazugehörige Workspace gezeigt. Der Artefakt-Browser zeigt den Inhalt eines ausgewählten Artefakts an. Eine besondere architektonische Herausforderung besteht bei diesem Ansatz darin, die verschiedenen Unternehmensanwendungen so in den Client zu integrieren, dass sie den Inhalt des Artefakts in dem Browser anzeigen können. Hierbei werden zwei Alternativen unterschieden: Entweder muss eine Adapter-Architektur geschaffen werden, zu der pro Anwendung ein Adapter entwickelt wird, der die Integration in den Client verantwortet. Alternativ dazu besteht die Integration darin, in die Erzeugung und Darstellung der Benutzeroberflächen der Anwendungen durch das Betriebssystem einzugreifen und sie so zu manipulieren, dass sie in dem Browser-Bereich „eingefangen“ werden. Aus Gründen der besseren Wartbarkeit ist die letzte Variante vorzuziehen,

da die Entwicklung und Pflege von zusätzlichen Adaptern überflüssig ist. Diese Variante wurde erfolgreich mithilfe eines Prototyps implementiert.

Ein Problem des oben beschriebenen Ansatzes ist, dass der Browser nicht flexibel genug ist. Zwar können dank eines Registerkarten-Konzepts mehrere Artefakte gleichzeitig geöffnet werden, aber die aktuellen vom Betriebssystem bereitgestellten Möglichkeiten zur Aufteilung von Fenstern auf dem Bildschirm werden durch dieses Konzept nicht unterstützt. Zudem führt das Einbinden der externen Anwendungen in den Browser in manchen Fällen zu unerwünschten Seiteneffekten wie zum Beispiel ausgeblendeten Menüs oder Pixel-Artefakten.

Ein anderer Ansatz verfolgt das Ziel, die Integration der externen Anwendungen zugunsten einer flexibleren und robusteren Darstellung der Artefakte aufzugeben. In diesem Ansatz ist der Browser nicht mehr enthalten. Stattdessen werden die Artefakte in herkömmlicher Weise durch die externen Anwendungen selbst dargestellt. Auf diese Weise kann der Case Manager auf eine größere Arbeitsfläche zur Strukturierung der Daten auf dem Workspace zurückgreifen. Auch dieser Ansatz wurde implementiert und lief in den Tests deutlich performanter im Vergleich zum ersten Ansatz.

In beiden oben beschriebenen Ansätzen ist immer nur der Workspace sichtbar, der zu dem aktuell ausgewählten Agendaeintrag gehört. Sofern also der Case Manager nicht alle seine Artefakte in einem Workspace sammelt, sind nie alle Artefakte zugleich sichtbar. Je nachdem, welche Strategie der Case Manager bei der Strukturierung seiner Agenda verfolgt, kann das eine Beeinträchtigung darstellen. Sind die Agendaeinträge klar abgrenzbare Aktivitäten oder Meilensteine, benötigt der Case Manager nur Zugriff auf die Artefakte der aktuellen Aktivität. Handelt es sich bei den Einträgen aber um semantisch stark verwobene Begriffe, so muss er häufig zwischen den Workspaces wechseln und hat infolgedessen einen eingeschränkten Überblick über wichtige Artefakte eines Falls. Deshalb wurde im Rahmen einer Konzeptstudie (Flock 2014) nach weiteren Ansätzen gesucht, um die Übersicht über die Artefakte eines Falls zu verbessern. Im Ergebnis wurden vier Varianten konzipiert, von denen eine Variante ausgewählt und in den Prototyp integriert wurde.

Eine der Varianten orientiert sich an der Funktionalität einer Facettensuche. Hierzu werden die Agendaeinträge mit je einer Auswahlbox versehen, die vom Case Manager beliebig an- und abgewählt werden kann (siehe Abbildung 63, links). Auf diese Weise kann der Case Manager nicht nur auf den Workspace des aktuell ausgewählten Agendaeintrags zugreifen, sondern zugleich die

Artefakte anderer Workspaces hinzuschalten. In dem abgebildeten Beispiel sind alle Artefakte der Agendaeinträge E, M und m sichtbar. Alle Artefakte des Agendaeintrags H sind ausgeblendet.

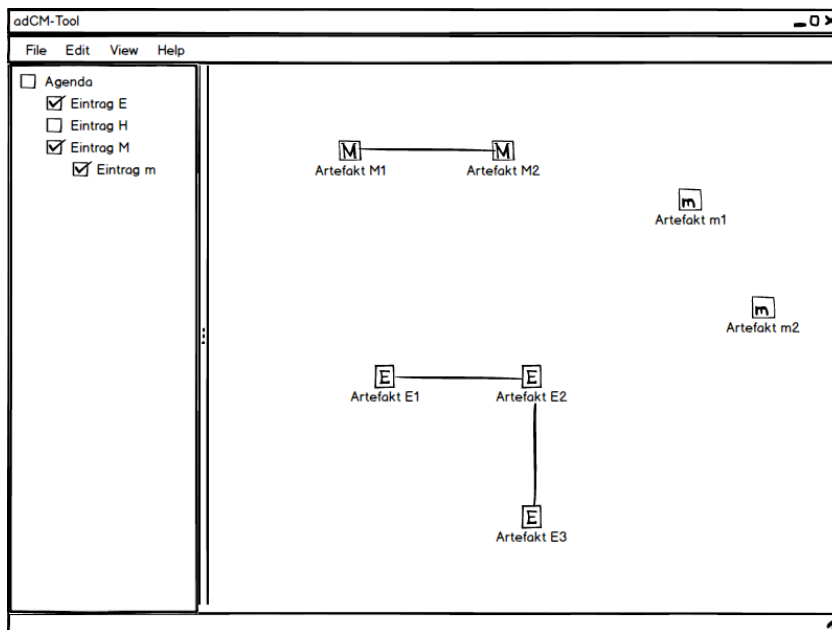


Abbildung 63: Integrierte Darstellung mehrerer Workspaces mithilfe von Auswahlboxen (Flock 2014)

Diese Alternative hat den großen Vorteil, dass auf diese Weise Artefakte agendaeintragsübergreifend miteinander in Beziehung gesetzt werden können. Allerdings muss der Case Manager bei dieser Variante stets berücksichtigen, dass es sich immer nur um eine individuell gefilterte Darstellung des gesamten Workspaces handelt. Er muss also zugleich immer die aktuelle Filtereinstellung beachten, um nicht versehentlich ein wichtiges Artefakt aus einem abgewählten Agendaeintrag zu übersehen.

Ein anderer Ansatz verfolgt ebenfalls das Ziel, eine bessere Übersicht durch die Integration von Workspaces in eine gemeinsame Darstellung zu erreichen. Gemäß dieser Variante werden neben den Artefakten des aktuell ausgewählten Agendaeintrags auch alle Artefakte der untergeordneten Agendaeinträge angezeigt. Das Beispiel auf Abbildung 64 zeigt alle Artefakte des ausgewählten Agendaeintrags M. Dazu zählen die unmittelbar diesem Eintrag zugeordneten Artefakte M1 und M2 sowie die Artefakte m1 und m2 des untergeordneten Eintrags m. Die Artefakte der Einträge E und H sind hingegen ausgeblendet.

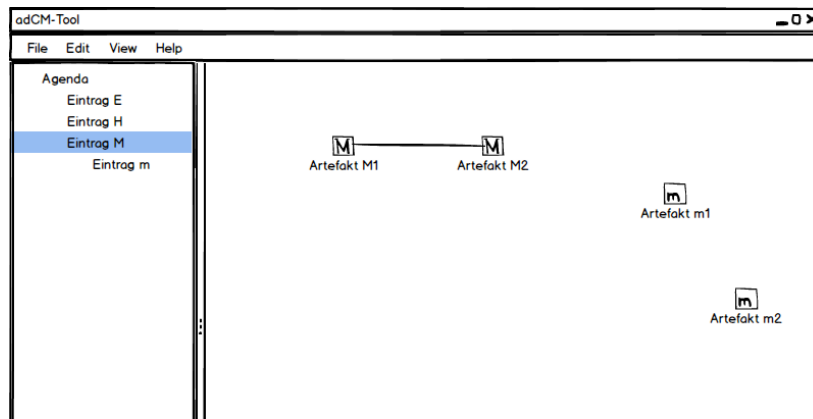


Abbildung 64: Integrierte Darstellung mehrerer Workspaces entlang ihrer hierarchischen Beziehung (Flock 2014)

Dieser Ansatz hat den Vorteil, dass durch nur einen Klick auf das Wurzelement alle Artefakte angezeigt werden können. Zudem ist es durch die Adaption des hierarchischen Konzepts für den Case Manager einfacher nachzuvollziehen, welche Auswahl an Artefakten er gerade auf dem Workspace sieht. Im Gegensatz zur zuletzt genannten Variante ist es also unwahrscheinlicher, dass Artefakte aufgrund einer falschen Filtereinstellung übersehen werden. Andererseits ist diese Variante weniger flexibel an die Informationsbedürfnisse des Case Managers anpassbar.

Ein großes Problem der bislang genannten Varianten ist, dass es bei einer Integration der Workspaces für den Case Manager nicht ersichtlich ist, welche Artefakte zu welcher Agenda gehören. Zwar könnte er sich diese Information durch das Wechseln zwischen, beziehungsweise das Ein- und Ausschalten der verschiedenen Workspaces erschließen, aber das ist offensichtlich mit einem gewissen kognitiven Aufwand verbunden, der die Aufmerksamkeit von den fachlichen Aufgaben ablenkt. Eine dritte Variante soll hier Abhilfe schaffen, indem sie alle Artefakte aller Workspaces zeitgleich anzeigt, jedoch die Artefakte des aktuell ausgewählten Agendaeintrags visuell hervorhebt. Bezogen auf das durchgehende Beispiel bedeutet das, dass zwar alle Artefakte der Einträge E, H, M und m dargestellt werden, aber nur die beiden Artefakte M1 und M2 des unmittelbar ausgewählten Eintrags (hier: Eintrag M) optisch hervorgehoben werden (siehe Abbildung 65).

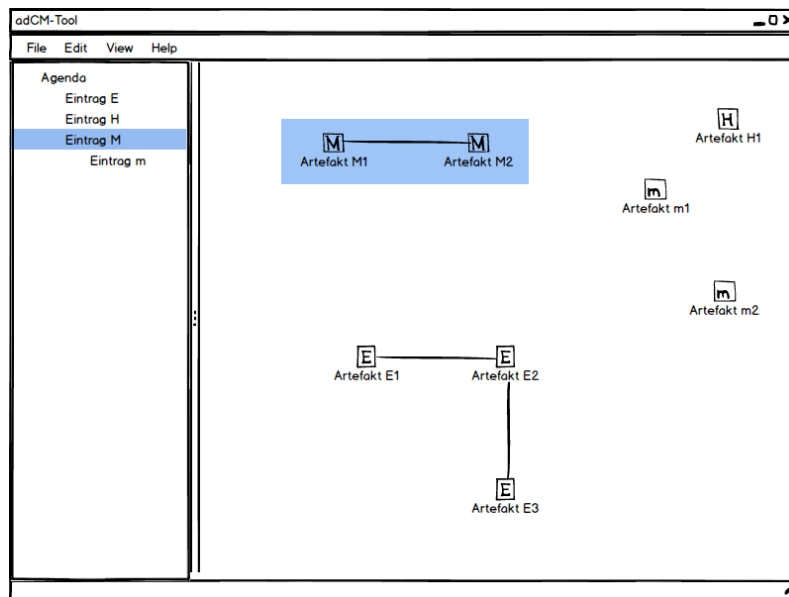


Abbildung 65: Zuordnung der Artefakte eines integrierten Workspace durch Hervorhebung (Flock 2014)

Zwar löst diese Variante das Problem der fehlenden Zuordnung von Artefakten eines integrierten Workspaces zu ihrem ursprünglichen Agendaeintrag. Dennoch ist dieser und allen übrigen bislang genannten Darstellungsformen gemein, dass die Art und Weise, wie die Artefakte auf dem integrierten Workspace platziert werden, unklar ist. Deshalb wurde eine weitere Variante entwickelt, die nicht nur alle Workspaces zugleich anzeigt, ohne dabei auch die Zuordnung zu den Agendaeinträgen zu vernachlässigen, sondern die auch ein Konzept zur Platzierung der Artefakte auf dem gemeinsamen Workspace umfasst. Sie greift auf das Treemap-Konzept zurück, einer speziellen Form der Darstellung für Elementen in einer hierarchischen Beziehung (Shneiderman 1992). Eine Treemap unterteilt die verfügbare Fläche rekursiv in Teilflächen, die im Falle einer Eltern-Kind-Beziehung übereinander und im Falle einer Vorgänger-Nachfolger-Beziehung nebeneinander angeordnet werden. Wie eine solche Unterteilung der Arbeitsfläche am bekannten Beispiel aussieht, zeigt Abbildung 66. Auf der untersten Ebene liegt die Fläche des Wurzelknotens. Sie füllt den verfügbaren Platz voll aus (am Beispiel ist das an dem Wurzelknoten der Agenda erkennbar). Die Flächen der Kindknoten liegen unmittelbar über dieser Fläche und teilen den verfügbaren Platz untereinander auf (am Beispiel an den Agendaeinträgen M, E und H). Dadurch sind die Flächen von Knoten mit gleicher Tiefe nebeneinander angeordnet.

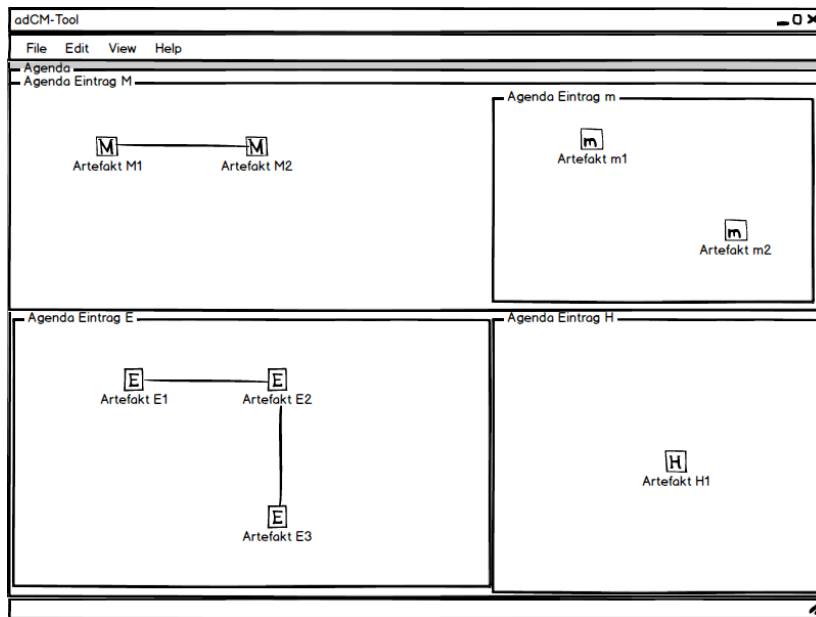


Abbildung 66: Integrierte Darstellung mehrerer Workspaces mithilfe einer Treemap (Flock 2014)

Ob diese Ordnung durch eine horizontale und vertikale Aufteilung des Platzes erfolgt, ist von der Implementierung des Treemap-Algorithmus abhängig. Die Algorithmen berücksichtigen dabei auch, dass die Flächen je nach Umfang des Inhalts (Agendaeinträge oder Artefakte) unterschiedlich groß sein müssen. Aus Platzgründen wird für weitere Details dazu auf die Literatur verwiesen (Shneiderman 1992; Bederson et al. 2002).

Die Design-Entscheidung für den Treemap-Algorithmus hat den Vorteil, dass die Agenda und die darin enthaltenen Workspaces gemeinsam abgebildet werden. Es ist also nicht mehr notwendig, die Agenda zusätzlich als Baumstruktur anzuzeigen. Außerdem hat dieser Ansatz den Effekt, dass das Auswählen eines Agendaeintrags nicht mehr notwendig ist, um einen Workspace sichtbar zu machen. Dadurch hat der Case Manager zwar einen besseren Überblick über die Artefakte in einem Fall, aber es kann nicht mehr so einfach vom System nachvollzogen werden, welcher Agendaeintrag gerade bearbeitet wird. Dieses Problem wurde gelöst, indem das Auswählen eines Artefakts zugleich den aktuellen Agendaeintrag auswählt. Auf diese Weise liefert das Arbeiten mit Artefakten bereits Informationen über die Abarbeitung der Agenda, ohne dass der Case Manager diese Informationen explizit preisgeben muss.

Weil der Treemap-Ansatz alle Artefakte eines Falls zugleich sichtbar macht, ist es im Vergleich zum ursprünglichen Ansatz aus (Benner et al. 2012) wahrscheinlicher, dass der Platz bei sehr vielen Artefakten nicht ausreicht. Deshalb wurde ein UI-Konzept entwickelt, das es dem Case Manager ermöglicht, für den aktuell bearbeiteten Agendaeintrag die gesamte zur Verfügung stehende Fläche auszunutzen. In der prototypischen Implementierung wurde diese Funktion mit

einem Doppelklick auf den Workspace realisiert. Wird auf dem vergrößerten Workspace erneut ein Doppelklick ausgeführt, kehrt der Case Manager in die ursprüngliche Darstellung mit allen Agendaeinträgen zurück.

7.6.4 UNTERSTÜTZUNG BEIM REFLEKTIEREN

Nachdem ein Case Manager relevante Artefakte gefunden und auf dem Workspace strukturiert hat, ist es im Rahmen der Arbeit mit diesen Artefakten eine zentrale Aufgabe, ihren Inhalt zu bewerten. Die (Zwischen-)Ergebnisse dieses Schritts können mithilfe von Annotationen festgehalten werden (siehe Kapitel 2.2.2). In (Benner-Wickner et al. 2013) wurden drei Herausforderungen identifiziert, die bei der Annotation von Artefakten auftreten. Sie wurden in Kapitel 4.1.4 als Anforderungen an das adCM-Konzept aufgenommen (Anforderungen 6-8) und im Szenario 6 in der Tabelle 17 konkretisiert. Demnach muss das adCM-Konzept eine Lösung dafür finden, wie

- a) Artefakte trotz fehlender Schreibrechte annotiert werden können,
- b) Annotationen unabhängig von der Art des Artefakts persistiert werden können und
- c) Inkonsistenzen in der Bedienung der Annotationen aufgelöst werden können.

Im weiteren Verlauf dieses Kapitels werden zunächst die Architekturentscheidungen begründet und im Anschluss die Details beschreiben, welche Technologien bei der Implementierung eingesetzt werden. Teile der Implementierung entstanden im Rahmen einer studentischen Ausarbeitung (Terziev 2013).

7.6.4.1 Architektur der Annotations-Dienste

Um die ersten beiden Anforderungen zu adressieren, sieht das adCM-Konzept einen separaten, zentralen Datenspeicher vor, in dem die Annotationen abgelegt werden. Zu jeder Annotation werden dabei auch Metadaten festgehalten, die unter anderem Aufschluss darüber geben, wo sich das annotierte Dokument befindet und an welcher Stelle im Dokument die Annotation positioniert ist.

Damit dieser Speicher für die Client-Komponenten des adCM-Frameworks erreichbar ist, wird er als Service in die Architektur integriert (siehe Abbildung 67). Über eine externe Verwaltungsschnittstelle können einzelne Annotationen neu hinzugefügt, geladen, aktualisiert und gespeichert werden. Diese Schnittstelle wird von verschiedenen Plug-Ins angesteuert, die die jeweilige Annotationsfunktion der übergeordneten Anwendung durch eine einheitliche Benutzerschnittstelle ersetzt (Anforderung 1). Darüber hinaus bietet der Service aber auch eine

Management-Schnittstelle, die über einen Browser zugreifbar ist. Darunter zählt das Registrieren von Benutzern – schließlich handelt es sich bei den Annotationen um Informationen, die nur dem jeweiligen Autor zugänglich sein dürfen. Außerdem kann über diese Management-Schnittstelle auch eine Übersicht über alle Annotationen erstellt werden, unabhängig zu welcher Art von Artefakt diese hinterlegt wurden.

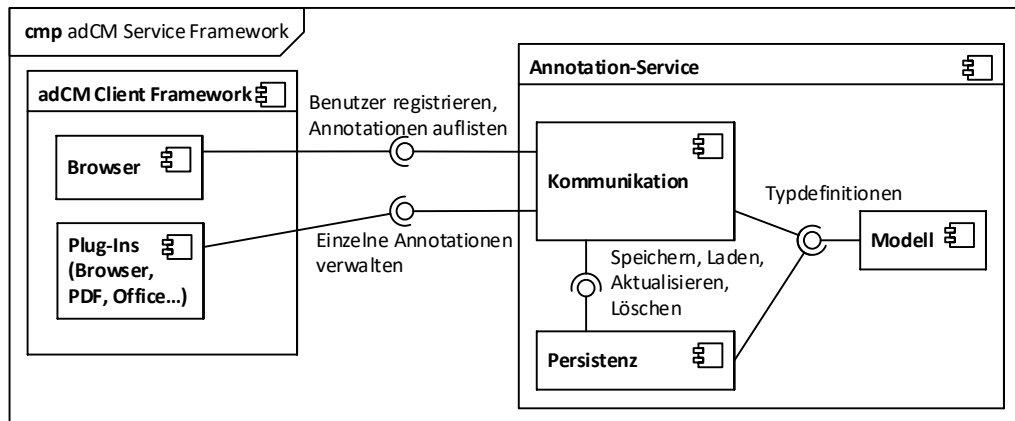


Abbildung 67: Architektur des Annotations-Service

Die Plug-Ins für die jeweiligen Anwendungen stellen sicher, dass die in Kapitel 2 definierte Grundfunktionalität zur Annotation von Artefakten genutzt werden kann. Das ist notwendig, weil die Annotationsfunktionalität nicht standardisiert ist (Benner-Wickner et al. 2013). Je nach Anwendung fallen die Möglichkeiten zur Annotation von Artefakten unterschiedlich aus. Hinzu kommt, dass es auch Anwendungen gibt, die im Kontext der adCM-Methode zwar einen Zugang zu wichtigen Artefakten darstellen, aber keine Annotationsfunktion bieten. Ein Beispiel hierfür ist der Browser. Er spielt eine wichtige Rolle beim Zugriff auf externe Datenquellen, bietet aber ohne Weiteres keine Annotationsfunktion.

Die Plug-Ins bestehen aus zwei Komponenten: einem Editor und einem Viewer (Benner-Wickner et al. 2013). Der Editor bietet eine Benutzerschnittstelle zur Erstellung und Bearbeitung von Annotationen. Damit können Inhaltspassagen des Artefakts mit Kommentaren, Schlagwörtern oder Bewertungen versehen werden. Die Funktionen dieser Komponente ergeben sich aus der Definition des Annotationsbegriffs und sind demnach unabhängig von der Anwendung beschreibbar. Dadurch kann eine anwendungsübergreifend einheitliche Benutzerschnittstelle für den Editor geschaffen werden. Der Viewer ist dafür verantwortlich, die erstellten Annotationen in der Anwendung darzustellen. Im Gegensatz zum Editor ist die Implementierung dieser Aufgabe stark von der Anwendung abhängig, weil zur Visualisierung der Annotationen sowohl in die Benutzeroberfläche der Anwendung als auch in die Datenstruktur des Artefakts eingegriffen

werden muss (Benner-Wickner et al. 2013). Details zu den konkreten technischen Herausforderungen werden im nachfolgenden Unterkapitel beschrieben.

7.6.4.2 Implementierung der Anwendungs-Plug-Ins

Um die Plug-Ins in die jeweiligen Anwendungen zu integrieren, werden – falls vorhanden – die anwendungsspezifischen Befehle zur Erstellung einer Annotation abgefangen und durch die Anzeige der oben beschriebenen Editor-Komponente ersetzt. Falls die Anwendung keine Annotationsfunktion anbietet, wird ein entsprechender Befehl hinzugefügt. Ein Beispiel hierfür ist der Browser. Um die Annotationsfunktion dort zu integrieren, wird das Plug-In in Form eines JavaScript-Programms in die Lesezeichenleiste eingebunden. Klickt der Benutzer auf das Lesezeichen, kann er fortan auf allen angesteuerten Webseiten markierte Textabschnitte mit einer Annotation versehen (siehe Abbildung 68).

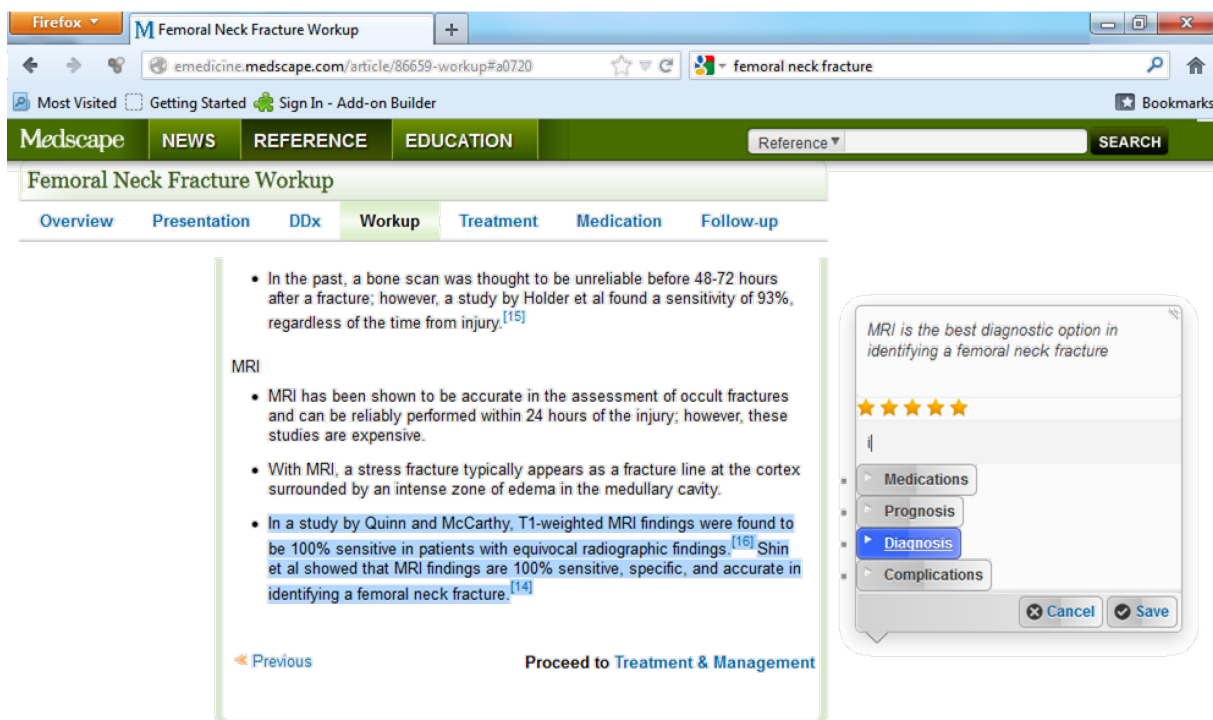


Abbildung 68: Integration der Annotationsfunktion im Browser (hier: Firefox)

Ein Beispiel für eine Anwendung, die im Gegensatz dazu bereits eine Annotationsfunktion anbietet, ist Microsoft Word. Hier wird der Befehl zur Erstellung eines Kommentars abgefangen und durch denselben Editor ersetzt, der auch im Browser zum Einsatz kommt. Zu diesem Zweck wird eine Webview-Komponente geöffnet, die über einen integrierten Browser verfügt und damit das JavaScript-Programm aufruft (siehe Abbildung 69).

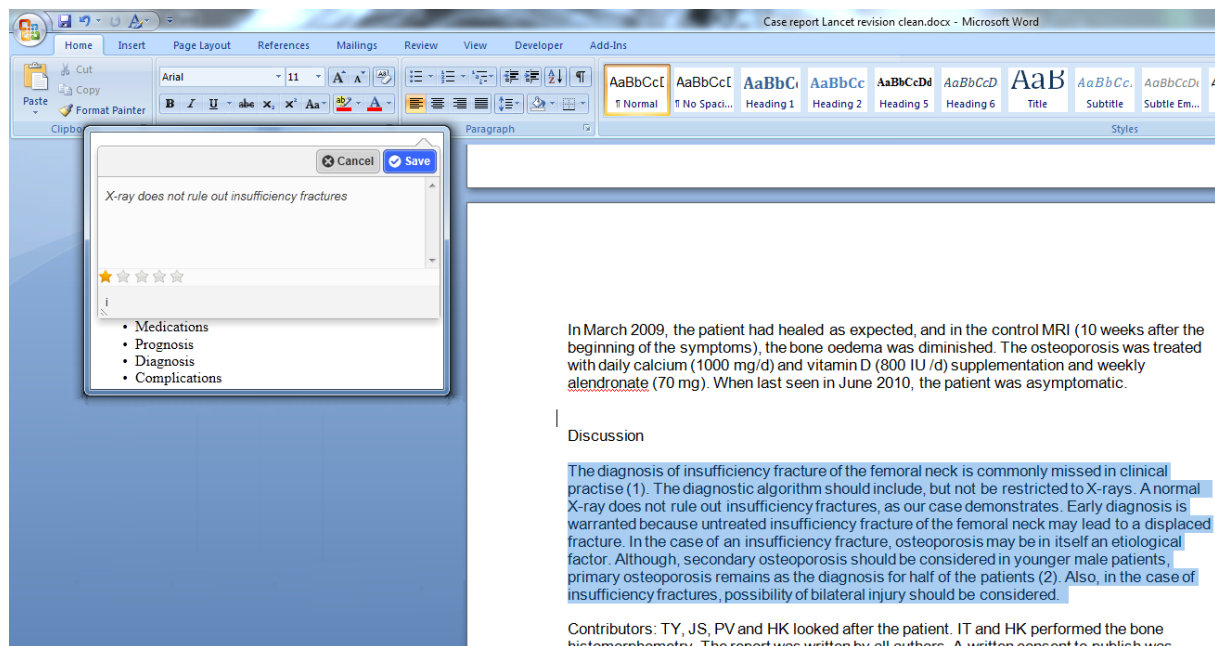


Abbildung 69: Integration der Annotationsfunktion in Microsoft Word

Die Verwendung desselben JavaScript-Programms stellt sicher, dass eine einheitliche Benutzerschnittstelle bereitgestellt wird und die Annotationsfunktionen anwendungsübergreifend zur Verfügung stehen. Zudem stellt die Viewer-Komponente der Plug-Ins sicher, dass beim Laden der Dokumente (beziehungsweise der Webseiten) die Annotationen vom Annotationsserver geladen und – falls vorhanden – mit den bereits existierenden anwendungsspezifischen Funktionen dargestellt werden. Das bedeutet am Beispiel von Microsoft Word, dass die Annotationen mithilfe der integrierten Kommentarfunktion angezeigt werden. Die über das Kommentieren hinausgehenden Funktionen (Bewertung und Verschlagwortung) sind in diesem Fall nur indirekt über den oben beschriebenen Aufruf des Editors zugänglich.

7.6.4.3 Implementierung des Annotations-Service

Weil es sich bei dem Annotations-Service um einen zustandslosen Dienst handelt, ist er – ebenso wie der Discovery-Service – als REST-Service implementiert. Um die Komplexität des Datenbankzugriffs zu reduzieren, wird JPA als Persistenzframework eingesetzt (Röder 2010). Dadurch reduziert sich der Programmumfang auf nur wenige Zeilen. Das Listing 10 zeigt, wie eine Annotation durch den Service gespeichert wird.

```
@Stateless
public class AnnotationRepository {
    @Inject
    private javax.persistence.EntityManager em;
    ...
    public Annotation save(Annotation annotation) {

        annotation = em.merge(annotation);
        return annotation;
    }
}
```

Listing 10: Reduktion der Codekomplexität durch Einsatz des JPA-Frameworks

Neben dem Zugriff auf die Annotationen per REST-Schnittstelle bietet der Annotation-Service auch eine Benutzerschnittstelle zur Darstellung der Annotationen an. Sie stellt die Annotationen in chronologischer Reihenfolge als HTML-Seite dar. Als Überschrift wird der Kommentartext verwendet, gefolgt von den Schlüsselwörtern und der annotierten Textpassage. Abschließend wird je Annotation ein Link auf das annotierte Artefakt angegeben.

Damit die Annotationen nur dem jeweiligen Autor zugänglich sind, ist für jedes Plug-In eine Authentifizierung erforderlich. Um zu vermeiden, dass sich der Case Manager für jede Anwendung einzeln anmelden muss, enthält die Architektur einen zentralisierten Authentifizierungsmechanismus. Hierzu wird mit dem Start des Desktop-Clients zugleich ein lokaler Proxy-Server gestartet, der als zentraler Zugriffspunkt für alle Plug-Ins dient. Nachdem sich der Case Manager authentifiziert hat, wird mithilfe eines Tokens (Jones et al. 2015) ein verschlüsselter Kanal zwischen dem lokalen Proxy und dem Authentifizierungsservice aufgebaut. Die Plug-Ins nutzen diesen Kanal, um ohne zusätzliche Authentifizierung die Annotationen laden zu können.

Mithilfe der prototypischen Implementierung des Annotation Service und den damit verbundenen Clients ist ein Konzept realisiert worden, das Case Manager dazu befähigt, Artefakte unabhängig ihres konkreten Typs zu annotieren. Zu diesem Zweck musste eine Lösung unter anderem dafür gefunden werden, dass der Case Manager auf manche Artefakte (zum Beispiel Links oder virtuelle Dokumente) keinen Schreibzugriff hat. Außerdem adressiert das Konzept das Problem, wie die Annotationen unabhängig vom jeweiligen Dateiformat und der damit einhergehenden unterschiedlichen Datenstruktur (HTML-, PDF- oder DOCX-Format) gespeichert werden können. Neben diesen technischen Herausforderungen adressiert das Konzept auch das Problem des unterschiedlichen Funktionsumfangs und der unterschiedlichen Bedienung der Annotationsfunktionen in den jeweiligen Office-Anwendungen.

7.7 RELATED WORK

Es gibt drei vergleichbare Forschungsvorhaben, die sich ebenfalls mit Werkzeugen zur Unterstützung von Case Managern auseinandergesetzt haben. Dabei wurden Architekturen entwickelt, gegen die sich die adCM-Architektur abgrenzen muss. In den folgenden Abschnitten werden diese Architekturen vorgestellt. Dabei wird kurz darauf eingegangen, wie die jeweilige Architektur mit den Inhalten der Dissertation in Bezug steht. Abschließend werden die Komponenten des adCM-Werkzeugs in eine übergeordnete Referenzarchitektur für Adaptive Case Management-Systeme eingeordnet.

7.7.1 ARCHITEKTUR DES ACTIVE KNOWLEDGE WORKSPACE

Im Rahmen des bis Anfang 2011 durch die EU geförderten Projekts „ACTIVE“ wurde unter anderem das Werkzeug „ACTIVE Knowledge Workspace“ entwickelt (Simperl et al. 2010). Es handelt sich dabei um eine Art Kontext-Sensor, der in das Betriebssystem und in Office-Anwendungen integriert wird. Der Sensor ersetzt nicht bestehende Anwendungen, sondern ist ein Hintergrunddienst, welcher in Zusammenarbeit mit Office-Plug-Ins den Arbeitskontext eines Case Managers anhand seiner Aktivitäten während der Bildschirmarbeit identifiziert. In Abhängigkeit dieses Kontextes werden Informationen gefiltert. Case Manager können Annotationen zu Dokumenten erstellen und untereinander austauschen. Außerdem wird eine Suchmaske bereitgestellt, mit der nicht nur das lokale Dateisystem, sondern auch die Annotationen durchsucht werden können. Die Architektur des ACTIVE Knowledge Workspace unterscheidet zwischen den oben genannten clientseitigen Komponenten einerseits und serverseitigen Infrastrukturkomponenten andererseits, die zentrale Aufgaben übernehmen wie zum Beispiel die Verwaltung von Metadaten oder auch das Text Mining. Die Architektur wird in (Ermolayev et al. 2008) und (Simperl et al. 2010) beschrieben und im weiteren Verlauf dargestellt.

Die strukturelle Anordnung der ACTIVE-Komponenten entspricht einer Client-Server-Architektur. Die Client-Komponenten sind für die Integration der serverseitig bereitgestellten Funktionen in eine Microsoft Windows-Plattform verantwortlich. Zu den Client-Komponenten zählen die **ACTIVE Taskbar** sowie **Plug-Ins** für verschiedene Anwendungen (zum Beispiel Office-Programme und Internet-Browser). Zu den Server-Komponenten zählen der Task Service, das Semantic Media Wiki und die ACTIVE Knowledge Workspace Services (siehe Abbildung 70). Die ACTIVE Taskbar besteht aus den drei Komponenten **Task Wizard**, **Task Pane** und **Recording**.

Die **Task Wizard** Komponente bietet dem Wissensarbeiter einen geführten Dialog, um Templates zu erstellen, zu betrachten oder mit anderen Wissensarbeitern zu teilen. Templates sind in diesem

Zusammenhang als wiederverwendbare Vorlagen mit einer vordefinierten Abfolge von Tasks definiert. Für die Ablage der Templates wird die Komponente **Semantic Media Wiki** verwendet (Krötzsch et al. 2007). Dabei handelt es sich um eine Erweiterung des frei verfügbaren Wiki-Systems MediaWiki, das auch die technologische Grundlage von Wikipedia bildet. Dessen Funktionalität wurde speziell auf die Zusammenarbeit unter Wissensarbeitern zugeschnitten, sodass mit dem System nicht nur Inhalte geteilt, sondern auch zum Zwecke der maschinellen Auswertung semantisch verknüpft werden können. Für die Integration des Semantic Media Wiki in die Architektur des ACTIVE Knowledge Workspace wurden die beiden Komponenten **Template Manager** und **Template Repository** ergänzt, um neben den textbasierten Inhalten auch die Templates verwalten und speichern zu können.

Die Komponente **Task Pane** ermöglicht dem Wissensarbeiter, eine neue Prozessinstanz für den zu bearbeitenden Fall zu erstellen und zu verwalten. Hierzu können bestehende Templates wiederverwendet werden. Zudem kann der Wissensarbeiter den Tasks einzelne Ressourcen (zum Beispiel Office-Dateien) zuordnen und direkt auf diese zugreifen. Mit der Komponente **Recording** werden die einzelnen Benutzerinteraktionen, zum Beispiel das Ansteuern einer bestimmten Internetseite, in Form von Ereignissen aufgezeichnet. Sie können vom Wissensarbeiter über die Task Pane den übergeordneten Tasks zugewiesen werden.

Die Komponente **Task Service** gehört zu den serverseitigen Komponenten der ACTIVE Knowledge Workspace Architektur. Ihre zentrale Aufgabe besteht darin, die aufgezeichneten Tasks und deren Ressourcen zu verwalten und zu persistieren. Sie stellt eine Schnittstelle zur Task Pane bereit, die die Benutzeroberfläche zur Verwaltung der Tasks darstellt. Zudem kommuniziert der Task Service mit einer Reihe von **Plug-Ins**, die in Windows-Anwendungen wie Office-Software oder Internet-Browsern integriert und für die Aufzeichnung von Tasks verantwortlich sind. Der Task Service kommuniziert darüber hinaus über eine Messaging-Infrastruktur mit den zentralen Diensten des ACTIVE Knowledge Workspace (**ACTIVE Knowledge Workspace Services**). Dazu zählen ein Dienst für das Vorschlagen von Ressourcen und Schlagworten (**Recommender**), ein Dienst für die Analyse vergangener Prozesse (**Task Mining**) sowie ein Dienst zur Erhebung des jeweiligen Arbeitskontextes eines Wissensarbeiters zur Laufzeit (**Context Mining**). Dieser Kontext kann aus Personen, Ressourcen und Tasks bestehen und wird dafür verwendet, bestimmte Funktionen kontextspezifisch anzubieten wie zum Beispiel das Filtern von Suchergebnissen oder Posteingängen.

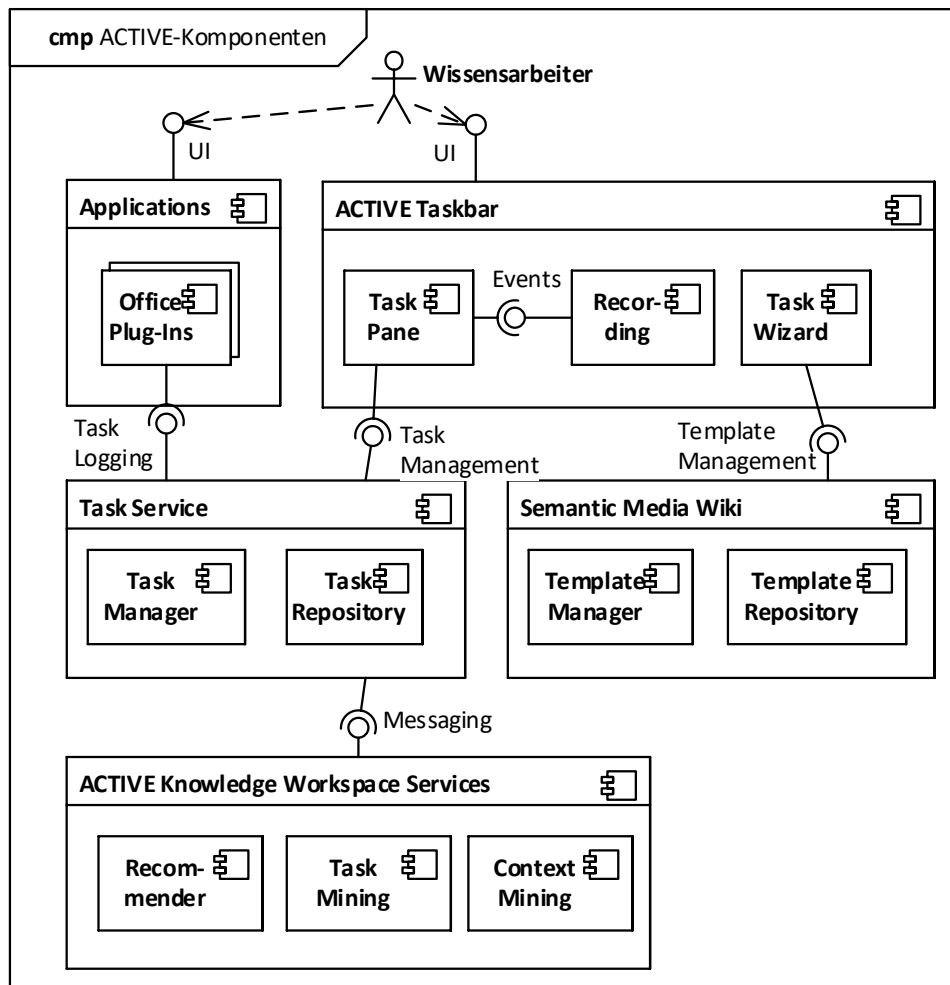


Abbildung 70: Architektur des ACTIVE Knowledge Workspace (UML-Komponentendiagramm)

Auf funktionaler Ebene weisen die im ACTIVE Knowledge Workspace eingesetzten Konzepte Ähnlichkeiten mit den adCM-Konzepten auf. In beiden Fällen werden Templates aus einzelnen Aufgaben eingesetzt, um den wissensintensiven Prozess zu unterstützen. Die Unterschiede werden erst bei einer detaillierten Betrachtung deutlich. So besteht ein wesentlicher konzeptueller Unterschied etwa darin, dass die Erfassung und Strukturierung dieser Aufgaben sowie deren Zuordnung zu Dokumenten im adCM-Ansatz bereits durch die Planung des Falls mithilfe der Agenda und des Workspace entstehen.

Die Ähnlichkeit der beiden Ansätze zeigt sich auch auf der Architekturebene. Beispielsweise sieht auch die adCM-Architektur vor, bestehende Office-Anwendungen um Plug-Ins zu erweitern, anstatt sie gänzlich durch eine eigene Implementierung zu ersetzen. Beide Architekturen sehen also vor, dass der Case Manager weiterhin die bestehenden Office-Anwendungen einsetzt, um den Geschäftsprozess zu bearbeiten. In beiden Architekturen erfüllen die Plug-Ins denselben Zweck:

Sie erfassen jeweils die Benutzerinteraktionen und integrieren zugleich spezifische Funktionen für Wissensarbeiter.

Darüber hinaus sieht die Architektur des adCM-Werkzeugs ebenfalls die Bereitstellung einer zentralen Komponente für eine Suchfunktion vor, um die Rechercheaktivitäten des Case Managers zu unterstützen. Allerdings werden die Suchergebnisse gemäß der adCM-Architektur nicht kontextbezogen gefiltert, sondern nur in ihrer Reihenfolge angepasst.

Beide Architekturen teilen die Verantwortlichkeiten sehr ähnlich auf client- und serverseitige Komponenten auf. Zum Beispiel wird das Mining in beiden Architekturen serverseitig bereitgestellt. Auch die zwischen den Wissensarbeitern geteilten Informationen in Form von Templates werden jeweils zentral abgelegt.

Verglichen mit der adCM-Architektur bietet der ACTIVE Knowledge Workspace mithilfe des Semantic Media Wikis auch eine Funktion, um beliebige andere Informationen über die Prozesse zu teilen. Die während der Nutzung dieser Komponente erstellten Tags werden im ACTIVE Knowledge Workspace dafür genutzt, eine Ontologie aufzubauen, mit der die Kontexte besser erkannt und unterschieden werden können. In der adCM-Architektur fällt das Sammeln von Schlagwörtern in die Verantwortung der separaten Annotationskomponente.

Auch bei der Kommunikation zwischen den einzelnen Komponenten werden in beiden Architekturen plattformunabhängige Messaging-Infrastrukturen eingesetzt, wobei der ACTIVE Knowledge Workspace die Software Apache ActiveMQ anstatt RabbitMQ verwendet. Im Unterschied zur adCM-Architektur wird dabei aber neben einfacher REST-Schnittstellen auch das SOAP-Protokoll (W3C Recommendation REC-soap12-part1-20070427) eingesetzt.

7.7.2 PROCESS MINING: PROCESS AWARE INFORMATION SYSTEMS

Im Process Mining nach (van der Aalst 2011) fallen Systeme, die ähnlich dem adCM-Werkzeug Sachbearbeiter zur Laufzeit bei der Bearbeitung von Geschäftsprozessen unterstützen, unter den Begriff „Process Aware Information System“ (PAIS). Um die Architektur solcher Systeme besser mit der des adCM-Werkzeugs vergleichen zu können, erfolgt im weiteren Verlauf ein tiefergehender Einstieg in die Architektur eines PAIS. Hierzu wird auf die einzelnen Bestandteile nach (Reichert und Weber 2012, 30ff) eingegangen (siehe Abbildung 71).

Ein PAIS besteht aus **Frontend**- und **Backend**-Komponenten. Das Backend enthält einen **Prozessmodell-Editor**, mit dem ein Prozess-Designer über eine Benutzerschnittstelle ein

ausführbares **Prozessmodell** erzeugen kann. Es enthält Aktivitäten, die einzelnen **Akteuren** (Mitarbeiter, Rollen, Gremien etc.) aus der Organisationsstruktur zugewiesen werden können.

Das Prozessmodell wird in eine **Prozess-Engine** geladen, die ebenfalls Bestandteil des Backends ist. Sie wird über eine UI-Schnittstelle von einem **Admin** konfiguriert. Ihre Verantwortung besteht unter anderem darin, alle automatisierbaren Aktivitäten mithilfe externer fachlicher Dienste (zum Beispiel **Webservices**) ausführen zu lassen. Handelt es sich um nicht-automatisierbare, manuelle Aktivitäten, werden die Aufgaben von der Prozess-Engine über eine **Aufgabenverwaltung** an die jeweils zugewiesenen Akteure delegiert. In beiden Fällen werden die Prozessausführungen für spätere Analyse Zwecke protokolliert.

Die Aufgabenverwaltung bildet das Frontend eines PAIS. Sie beauftragt einzelne **UI-Services** mit der Erstellung von Dialogmasken, die jeweils auf die Ausführung der manuellen Aktivitäten ausgerichtet sind. Zudem bietet sie eine Benutzerschnittstelle an, um den Akteuren die erzeugten Dialogmasken und Aufgabenlisten darzustellen.

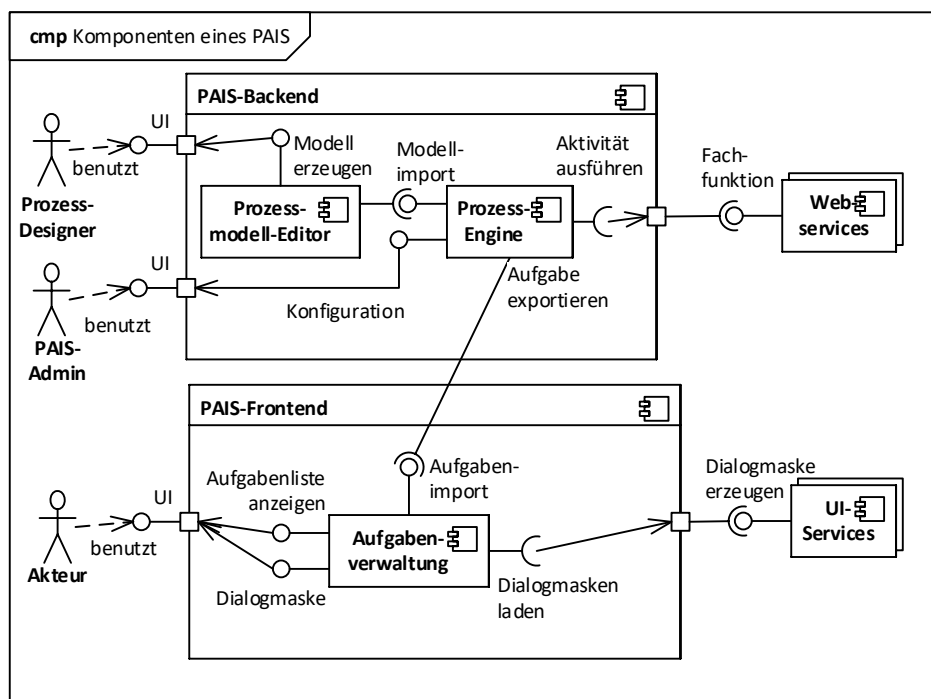


Abbildung 71: Komponenten eines PAIS nach (Reichert und Weber 2012, S. 32) als UML-Komponentendiagramm

Die PAIS-Architektur ähnelt der adCM-Architektur nicht nur insofern, als dass Akteure der Prozesse in beiden Fällen mit einer Komponente arbeiten, die Ihnen eine Liste von Aufgaben präsentiert. In beiden Fällen sind darüber hinaus die Aufgaben mit einem Modell verknüpft, das in serverseitigen Komponenten erzeugt und verwaltet wird. Allerdings unterscheiden sich die

Modelle wesentlich in ihrer Struktur. Denn bei der PAIS-Architektur handelt es sich um Modelle strukturierter Geschäftsprozesse mit Ketten aus teils automatisierbaren und nicht-automatisierbaren Aktivitäten. Der adCM-Ansatz sieht stattdessen Modelle unstrukturierter Geschäftsprozesse mit Hierarchien von rein manuellen Aktivitäten, denen keine Reihenfolge streng vorgegeben ist.

Um die tatsächlich ausgeführten Aktivitäten den in den Modellen vorgesehenen Aktivitäten gegenüberstellen zu können, sieht sowohl die PAIS- als auch die adCM-Architektur eine serverseitige Monitoring-Komponente zum Protokollieren der Aktivitäten vor. In der adCM-Architektur ist darüber hinaus aber auch eine Discovery-Komponente vorgesehen, die diese Protokolle als Datenbasis für die Erzeugung von Modellen verwenden. Für eine solche Funktionalität muss ein PAIS erst mit einer separaten Discovery-Komponente ausgestattet werden. Wie eine solche Erweiterung der Architektur gestaltet werden kann, wird in (van der Aalst et al. 2010) beschrieben. Dabei handelt es sich um eine Beispiel-Architektur, mit der ein PAIS an das erweiterbare Process Mining-Werkzeug ProM angebunden werden kann (siehe Abbildung 72). Die Komponente **ProM** ist eine Java-Anwendung und besteht aus Unterkomponenten in Form von Plug-Ins, die verschiedene Algorithmen anbieten und individuell zur Laufzeit nachgeladen werden können. Das vom Prozess-Analyst ausgewählte **Process Discovery** Plug-In übernimmt dann die Aufgabe, die vom PAIS erzeugten Ereignisprotokolle zu analysieren. Aus dieser Analyse resultieren je nach Algorithmus unterschiedliche **Prozessmodelle**. Sie werden zur Laufzeit von den ausgewählten **Recommendation** Plug-Ins herangezogen, um den Benutzer via PAIS mit Informationen über den Prozess zu versorgen.

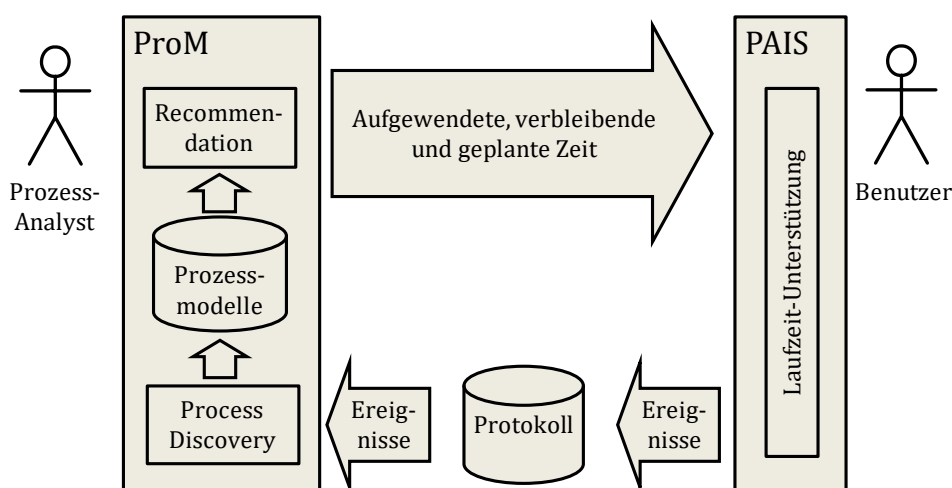


Abbildung 72: Integration von Unterstützungssystemen im Process Mining-Umfeld nach (van der Aalst et al. 2010)

Eine sehr ähnliche Aufteilung findet auch in der Architektur des adCM-Werkzeugs statt: Die Funktionen zur Laufzeit-Unterstützung werden von einem Client bereitgestellt, der vom Case Manager bedient wird. In diesem Sinne übernimmt der Client die Rolle des PAIS. Die dafür erforderlichen Informationen werden aus den protokollierten Ereignissen gewonnen, die von den serverseitigen Komponenten ausgewertet werden. Diese Komponenten erfüllen demnach die Rolle von ProM.

7.7.3 DYONIPOS-PROJEKT

Ein weiteres Werkzeug zur proaktiven Unterstützung von Wissensarbeitern wurde im Rahmen des DYONIPOS-Projekts entwickelt (Makolm et al. 2008). Dessen Zielsetzung ist die an den Bedarf des Wissensarbeiters ausgerichtete, kontextsensitive Filterung und Bereitstellung von Informationen während der Ausführung kollaborativer, wissensintensiver Geschäftsprozesse. Ähnlich dem in dieser Dissertation vorgestellten Konzept beobachtet das im DYONIPOS-Projekt entwickelte Werkzeug zu diesem Zweck den Wissensarbeiter bei seinen Tätigkeiten. Anhand der Aufzeichnungen erkennt es den Kontext der Tätigkeit. Die Grundlage hierfür bildet die unter anderem von Andreas Rath entwickelte Methode zur ontologiebasierten Identifikation des Interaktionskontextes (Rath 2010). Die Architektur des Werkzeugs wird in (Rath 2007) detailliert beschrieben und ist in Abbildung 73 dargestellt.

Für die Aufzeichnung der Benutzerinteraktionen kommt eine in C# entwickelte Komponente mit dem Namen **Context Observer** zum Einsatz. Im Sinne einer Client-Server-Architektur aggregiert sie Ereignisse, die von einzelnen Sensoren aufgezeichnet wurden. Als Sensoren kommen einzelne Plug-Ins für Office-Anwendungen (**Office Sensors**) und Internet-Browser (**Browser Sensors**) zum Einsatz sowie ein **generischer Sensor**, der relevante Ereignisse aus dem Betriebssystem abfängt. Für die Übertragung der Ereignisse sieht die Architektur ein XML-Format vor.

Die gesammelten Ereignisse werden vom Context Observer an die Java-basierte Komponente **Task Recognizer** weitergereicht. In zwei Schritten erzeugt sie aus den einzelnen technischen Ereignissen (zum Beispiel Mausklicks und Tastatureingaben) fachliche Aktivitäten (Tasks). Für jeden Schritt sieht die Architektur eine separate Komponente vor: den **Event Block Generator**, der mithilfe einer fachlichen Regelbasis zusammenhängende Ereignisse zu Blöcken zusammenfügt sowie ein **Task Generator**, der den Blöcken Aktivitäten zuweist. Hierzu wird auf das **Know Miner Framework** zurückgegriffen, einem externen Werkzeug, das Dienste zum Gruppieren von Elementen anbietet. Dieses Werkzeug dient darüber hinaus dem Task Recognizer dazu, passende Ressourcen für einen Task zu finden. Hierzu ist es an die Volltext-Suchmaschine **Apache Lucene** angeschlossen.

Alle Ereignisse, Ereignisblöcke und Tasks werden als Ontologie in einer lokalen Wissensdatenbank (**Personal Knowledge Base**) gespeichert. Hierzu wird ein semantisches Datenmodell verwendet, in dem die einzelnen Einträge dynamisch miteinander in Beziehung gesetzt werden können. Sie werden als Tripel in der Sprache RDF gespeichert.

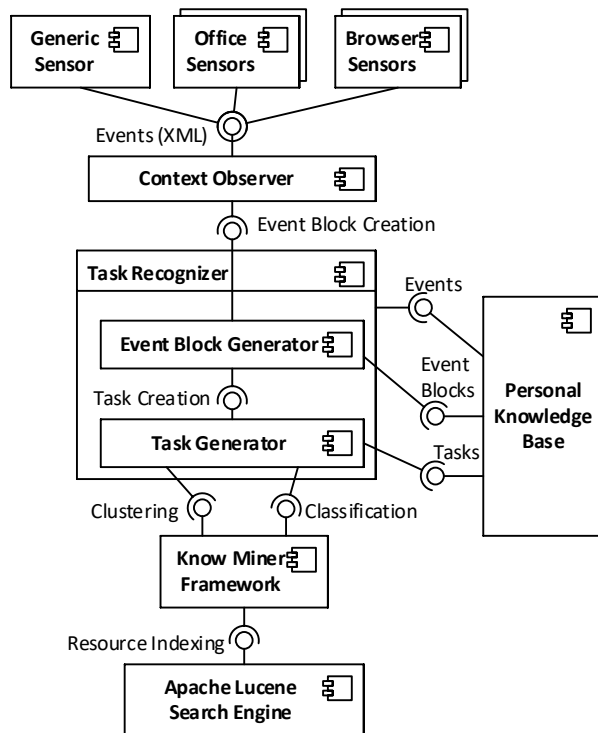


Abbildung 73: Komponenten der DYONIPOS Architektur nach (Rath 2007) als UML Komponentendiagramm

Die DYONIPOS-Architektur besitzt im Vergleich zur adCM-Architektur einen sehr ähnlichen Aufbau. So werden auch hier Plug-Ins für Windows-Anwendungen eingesetzt, die mit einer zentralen Komponente kommunizieren. Ebenfalls werden in beiden Architekturen die Benutzerinteraktionen aufgezeichnet und zentral ausgewertet. Im Gegensatz zur adCM-Architektur werden die Ereignisse von den Sensoren allerdings zunächst auf einer sehr technischen Ebene aufgezeichnet, die sich nicht ohne Weiteres fachlichen Aktivitäten zuordnen lassen. Die dafür erforderlichen Schritte sind die Kernfunktionalität des Task Recognizers, der zentralen Komponente der DYONIPOS-Architektur.

Im Vergleich zum adCM-Werkzeug konzentriert sich das DYONIPOS-Werkzeug auf das Modellieren einer Wissensbasis und der kontextbezogen gefilterten Bereitstellung von Ressourcen. Es sieht keine Komponente zum Vorschlagen konkreter Aktivitäten vor. Allerdings verwenden beide Architekturen mit einem RDF Triple Store die gleiche Technologie zur Speicherung der Ereignisse und der daraus entwickelten Modelle. Ebenso sehen beide Architekturen die Integration einer Volltext-Suchmaschine vor.

7.7.4 EINORDNUNG IN EINE ACM-BEISPIELARCHITEKTUR

In (Pucher 2010) wird eine Architektur für ACM-Systeme beschrieben, die ebenfalls sowohl einen Schwerpunkt auf die Integration mit anderen Informationssystemen legt als auch die Anbindung externer Datenquellen explizit berücksichtigt. Dem Autor zufolge ist, im Gegensatz zu klassischen WfMS, eine solche Integration erforderlich, um die verschiedenen Facetten von Wissensarbeit durch ein Werkzeug zu unterstützen. Die meisten Komponenten der Architektur erfüllen Funktionen für die Kommunikation mit Partnern oder Kunden. Dazu zählen beispielsweise ein digitaler Posteingang mit Texterkennungsfunktion und Routing-Funktionalität zur automatischen Verteilung von Dokumenten, aber auch eine Textverarbeitungskomponente mit verschiedenen Möglichkeiten zur strukturellen Aufbereitung von ausgehenden Schreiben. Für die Unterstützung des Case Managements ist ebenfalls eine separate Komponente vorgesehen. Sie bietet sehr ähnliche Funktionen wie das in diesem Kapitel beschriebene Werkzeug. Dazu zählen das Erfassen von Aufgaben, eine Suchfunktion sowie die Protokollierung von Ereignissen.

In Abgrenzung zum adCM-Ansatz konzentriert sich diese Architektur auf die Unterstützung des Schriftverkehrs zwischen dem Case Manager und allen anderen Prozessbeteiligten. Eine automatische Analyse vergangener Fälle mit dem Ziel, potentiell relevante zukünftige Handlungsschritte zu identifizieren, ist nicht vorgesehen.

7.8 ZUSAMMENFASSUNG

In diesem Kapitel wurde anhand einer Beispiel-Architektur beleuchtet, wie die adCM-Konzepte im Zusammenspiel mithilfe eines integrierten Werkzeugs umgesetzt werden können und welche Technologien dabei zum Einsatz kommen. Entlang der übergeordneten softwaretechnischen Architektur wurden die Entwurfs- und Design-Entscheidungen der einzelnen Komponenten beschrieben.

Nach dem Prinzip der losen Kopplung trennt die übergeordnete Architektur die Client-Komponenten von den Services mithilfe standardisierter Web-Technologien. Dazu zählen die Protokollierung von Ereignissen über eine Message-Oriented-Middleware und die Anbindung des Discovery-Algorithmus als Webservice.

Zusätzlich zu diesen strukturbezogenen Entwurfsentscheidungen sind auch Entscheidungen vorgestellt worden, die sich auf das Verhalten des Systems beziehen. Hierzu zählt die Beschreibung eines 3-stufigen Algorithmus für die Implementierung einer Vorschlagsfunktion.

Damit wurde der adCM-Lebenszyklus vom Monitoring über das Discovery bis zum Vorschlagen von Templates geschlossen.

Neben tieferen Einblicken in die Umsetzung zentraler Konzepte wie dem Monitoring und dem Discovery wurde das Kapitel auch dafür genutzt, komplementäre Konzepte zu beleuchten. Damit sind vor allem die Design- und Implementierungsentscheidungen gemeint, die den Case Manager bei der Ausübung weiterer Kernaktivitäten wie dem „Recherchieren“, „Strukturieren“ und „Interpretieren“ unterstützten.

Zuletzt wurde die adCM-Architektur mit anderen Architekturen aus Forschung und Praxis in Bezug gesetzt. Dabei wurde deutlich, dass sich vereinzelte grundlegende Architekturentscheidungen auch in anderen Architekturen wiederfinden. Hierdurch wird der Eindruck gefestigt, dass die adCM-Architektur auf allgemein anerkannte Technologien aufsetzt. Weil sich die zugrundeliegenden konzeptuellen Ansätze aber in einzelnen Punkten jeweils unterscheiden, konnten auch einige Unterschiede in den Architekturen der Werkzeuge identifiziert werden. So sind beim adCM-Ansatz beispielsweise keine Komponenten für die Aggregation von technischen zu fachlichen Ereignisse erforderlich.

Für das nachfolgende Fallbeispiel (siehe Kapitel 8) wurden alle vorgestellten Komponenten der adCM-Architektur als Werkzeug prototypisch implementiert.

8 FALLBEISPIEL

Die in dieser Dissertation vorgestellten Konzepte zur Unterstützung von Case Managern in schwach strukturierten Prozessen wurden bislang nur einzeln untersucht. In diesem Kapitel soll entlang eines Beispiels gezeigt werden, wie sie im Zusammenspiel einen Case Manager bei der Fallsteuerung unterstützen. Hierzu wurde ein Fallbeispiel entworfen, das jeweils einen Anwendungsfall für die einzelnen Unterstützungsfunktionen berücksichtigt. Es wurde mit einer prototypischen Implementierung des in Kapitel 7 beschriebenen Werkzeugs durchgeführt und stellt in diesem Sinne ein Proof of Concept der adCM-Konzepte dar.

Das Kapitel beginnt mit einer kurzen Beschreibung des Designs dieser Untersuchung. Darin wird geklärt, um welches Beispiel es sich handelt, nach welchen Kriterien es ausgewählt wurde und welche Werkzeuge für dessen Vorbereitung und Durchführung zum Einsatz gekommen sind. Es folgt eine ausführliche Darlegung des fachlichen Kontexts, in dem sich das Beispiel befindet. Anschließend wird entlang bestimmter Zeitpunkte nachverfolgt, wie die Konzepte des adCM jeweils zum Einsatz kommen.

8.1 VORBEMERKUNGEN

Die Evaluation der integrierten Lösungsbausteine anhand einer Fallstudie aus der Praxis unterliegt einigen Herausforderungen. Zum einen muss eine möglichst große Fallbasis vorliegen. Weil das Case Management – wie in den einleitenden Kapiteln dargelegt – vor allem durch die intensive Arbeit mit Menschen gekennzeichnet ist, handelt es sich dabei in der Regel um personenbezogene Daten. Im deutschen Rechtsraum muss die bereitgestellte Fallbasis also entweder so anonymisiert/pseudonymisiert sein, dass sie keine Rückschlüsse auf die betroffenen Personen erlaubt, oder es müssen vertragliche Rahmenbedingungen geschaffen werden, in denen die Fallbasis im Sinne einer Auftragsdatenverarbeitung untersucht werden kann. Um eine solche Fallstudie vorzubereiten, wurden im Verlauf der Promotion verschiedene Unternehmen und Institutionen kontaktiert. Das Ergebnis dieser Anstrengungen war eine Kooperation mit einem Träger der deutschen gesetzlichen Unfallversicherung. Aus dieser Zusammenarbeit ergaben sich wertvolle Einblicke in die Praxis des Case Managements allgemein und des Reha Managements

insbesondere. Die Fallakten des Trägers bestanden jedoch nur aus digitalisierten, handschriftlichen Dokumenten, für deren manuelle Anonymisierung oder Pseudonymisierung keine Ressourcen verfügbar waren. Alternativ wurde deshalb ein Datenschutz- und Sicherheitskonzept nach § 9 BDSG sowie eine Datenschutzvereinbarung abgestimmt, um die Fallbasis im Originalzustand zu untersuchen. Allerdings hat das Bundesversicherungsamt in seiner Verantwortung als Rechtsaufsicht nicht die für die Bereitstellung der Daten erforderliche Zustimmung erteilt. Es werden zwar weitere Anstrengungen unternommen, um eine solche oder vergleichbare Evaluation in der Praxis durchzuführen. Sie sind aber nicht mehr Bestandteil dieser Ausarbeitung.

Um dennoch zumindest im Sinne eines Proof of Concept einen Eindruck davon zu erhalten, wie die einzelnen Konzepte den Case Manager in einem konkreten praxisnahen Beispiel unterstützen können, wurde ein entsprechendes Fallbeispiel entworfen. Es berücksichtigt ausdrücklich einige Anwendungsfälle für die Unterstützung des Case Managers, etwa der Externalisierung und Fortentwicklung seines Plans.

Weil das Fallbeispiel bewusst so gewählt wurde, dass die Konzepte zur Unterstützung des Case Managers potentiell greifen können, wird am Ende des Kapitels eine kurze Diskussion darüber angeschlossen, in welchen Fällen die Konzepte *nicht* erfolgreich sind.

8.2 EINFÜHRUNG IN DAS FALLBEISPIEL

Das Beispiel greift thematisch das in Kapitel 4.3.4 eingeführte Fallbeispiel aus dem Reha-Management der gesetzlichen Unfallversicherung auf. Darin ist es die Aufgabe des Case Managers Herrn Müller, die individuelle Therapie des Patienten Herrn Schmidt zu begleiten und zu steuern. Für eine bessere Übersicht werden die für den Einsatz der adCM-Konzepte wesentlichen Ereignisse und Aktivitäten, die im Rahmen des Beispiels auftreten, entlang eines Zeitstrahls dargestellt (Abbildung 74). Die Unterbrechungen des Zeitstrahls sollen einen längeren Zwischenzeitraum darstellen, dessen Ereignisse für das Fallbeispiel irrelevant sind und deshalb nicht im Detail betrachtet werden.

Das bekannte Fallbeispiel wurde um ein Ereignis ergänzt, das vor dem Arbeitsbeginn von Herrn Müller stattfand: die regelmäßige Aktualisierung der Templates für den Reha-Management-Prozess. Dabei wurde die Fallbasis mithilfe des adCM Miners auf häufig wiederkehrende Agendaeinträge untersucht. Dieses und alle folgenden Ereignisse werden im weiteren Verlauf des Kapitels einzeln beschrieben. Dabei wird darauf eingegangen, wie die einzelnen adCM-Lösungsbausteine zur Unterstützung des Case Managers beitragen.

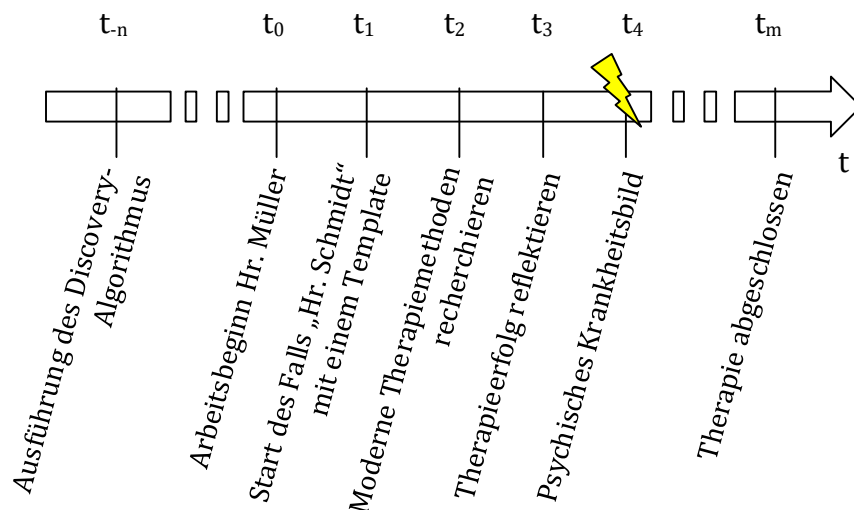


Abbildung 74: Verlauf des Fallbeispiels entlang der Zeitachse mit unvorhergesehenem Ereignis (Blitz)

Um die erforderliche Fallbasis unter praxisnahen Bedingungen aufzubauen, wurden die Daten nicht separat durch einen Generator erzeugt und direkt in die Datenbank geladen. Stattdessen wurde eine Software für automatisierte Benutzerakzeptanztests eingesetzt, die die Mausbewegungen und Tastatureingaben eines Case Managers für den Aufbau der Fallbasis simuliert. Auf diese Weise sollte erreicht werden, dass die Fälle mit derselben Benutzerschnittstelle erstellt und bearbeitet werden, die auch in der Praxis den Case Managern zur Verfügung stehen würde. Zwar können auf diese Weise die unterschiedlichen Arbeitsweisen der Case Manager nicht realistisch berücksichtigt werden. Trotzdem wurde aber zumindest darauf geachtet, dass jeder Fall aufgrund der individuellen Heilverläufe der Patienten verschieden ist und nur die im Abschnitt 8.3 beschriebenen Teile der jeweiligen Agenda übereinstimmen.

8.3 BESCHREIBUNG DER FALLBASIS

Die Fallbasis wurde so aufgebaut, dass es sich bei dem Beispiel nicht um den ersten Fall handelt, der in der Reha-Management-Abteilung durchgeführt wurde. Zum Zeitpunkt t_n (Ausführung des Discovery-Algorithmus, siehe 8.4.1) sind bereits 200 Fälle mit dem adCM-Werkzeug bearbeitet worden. Alle Fälle unterscheiden sich zwar mindestens in einem Eintrag in der Agenda. Aber weil sich manche Case Manager-Kollegen von Herrn Müller über die grundlegenden Phasen eines Falls im Reha Management austauschen und deshalb ihre Arbeit auf abstrakter Ebene ähnlich strukturieren, wirken sich diese Unterschiede nicht auf einen stets wiederkehrenden Teil der Agenda aus. Die Agenden von 120 Fällen – das sind 60% der Fallbasis – sind demnach bezogen auf einen Teilbaum aus fünf Knoten identisch (siehe Abbildung 75). Diese fünf Knoten

repräsentieren die von der Reha-Management-Abteilung empfohlenen wesentlichen Case Management-Grundschrte.

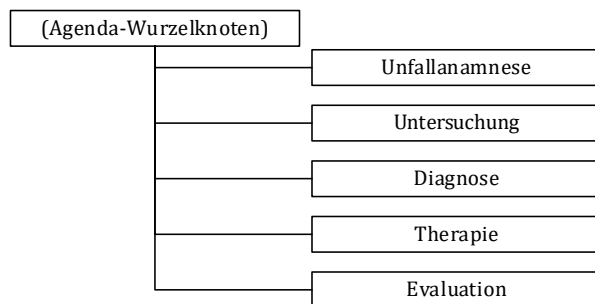


Abbildung 75: Gliederung der Agenda, die auf 60% der Fälle zutrifft

Neben diesen grundlegenden Schritten in der ersten Ebene der Agenda gibt es auch ein häufig beobachtbares Muster, das sich auf einzelne Abschnitte in tieferen Gliederungsebenen bezieht. Das Muster kann in etwas mehr als 40% der Fälle ermittelt werden und entsteht bei der Steuerung von Fällen, in denen plötzlich psychische Unfallfolgen, genauer gesagt posttraumatische Belastungsstörungen (PTBS), diagnostiziert werden. Solche Störungen treten bei schweren Unfällen sehr häufig auf, werden aber oft erst Monate später erkannt. Das in der Agenda wiederkehrende Muster (schematisch dargestellt in Abbildung 76) enthält in der ersten Ebene zunächst fast alle der häufig wiederkehrenden Grundschrte. Darüber hinaus enthält es aber auch drei typische Untersuchungsaspekte und eine etablierte Therapiestruktur, die im Falle einer PTBS (ICD-Code: F 43.1) von den Kollegen von Herrn Müller stets explizit notiert wurden.

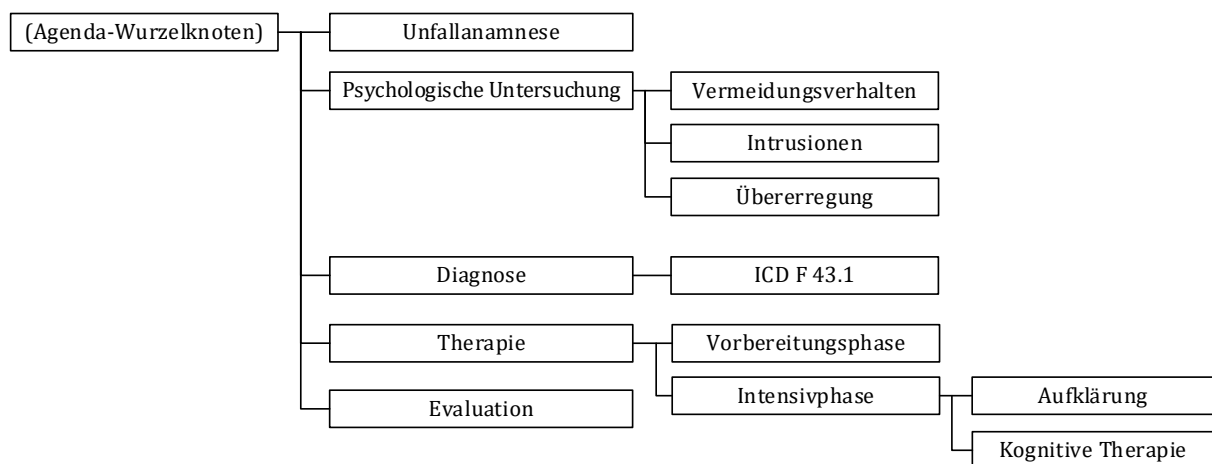


Abbildung 76: Wiederkehrendes Muster bei der Behandlung posttraumatischer Belastungsstörungen

Die Aufzählung der Schritte während der Intensivphase ist nicht vollständig. In sehr vielen Fällen (knapp 40% der Fallbasis) wurde zusätzlich die Exposition als Teil der Intensivphase betrachtet und medikamentös behandelt.

8.4 ABLAUF DES BEISPIELS

8.4.1 *T_N: AUSFÜHRUNG DES DISCOVERY-ALGORITHMUS*

Zum oben genannten Zeitpunkt wird der in Kapitel 6.3.4 beschriebene Discovery-Algorithmus gestartet. Der für die IT-Governance des Reha Managements verantwortliche Mitarbeiter hat den Algorithmus so einstellen lassen, dass die gesuchten Muster mindestens in 40% der Agenden in der Fallbasis vorkommen müssen. Bei einer Fallbasis aus 200 Agenden betrifft das also alle Agenda-Teilbäume, die in wenigstens 80 Agenden wiedergefunden werden können.

Der Algorithmus findet zunächst die fünf in Abbildung 75 gezeigten Grundschriffe, denn wie eingangs in der Ausgangssituation beschrieben wurde, treten diese Grundschriffe in 120 Fällen auf (60%). Zusätzlich zu diesem Muster findet er auch die oben beschriebenen Gemeinsamkeiten bei der Bearbeitung posttraumatischer Belastungsstörungen (siehe Abbildung 76). Er errechnet, dass dieses Muster in 80 Fällen erkannt werden kann und damit knapp die erforderliche Schwelle von 40% erreicht. Bei dem Versuch, dieses Muster im Sinne eines maximalen Teilbaums weiter zu vergrößern, berücksichtigt er unter anderem auch den zusätzlichen Agendaeintrag „Exposition“ in der Intensivphase. Dabei stellt der Algorithmus aber fest, dass der Teilbaum unter Berücksichtigung dieses Eintrags nur noch in 79 Fällen beobachtet werden kann. Damit liegt die Häufigkeit des ergänzten Teilbaums knapp unter der vom Prozessverantwortlichen festgelegten Schwelle. Deshalb, und weil kein anderer Agendaeintrag mehr gefunden werden kann, der zu einem größeren Teilbaum führt, ohne unter die geforderte Schwelle zu fallen, verwirft der Algorithmus diese Ergänzung.

Im letzten Schritt speichert der Algorithmus die beiden identifizierten maximalen Teilbäume jeweils in Form eines Template in der Graphdatenbank ab.

8.4.2 *T₀: ARBEITSBEGINN HERR MÜLLER*

In dem Fallbeispiel beginnt Herr Müller morgens gegen acht Uhr seinen Arbeitstag und findet in seinem Posteingang die Nachricht einer Vertragsklinik. Darin wird mitgeteilt, dass die im Heilverlaufsplan festgehaltene Therapie eines Patienten ab nächster Woche durchgeführt werden kann. Bei dem Patienten handelt es sich um Herrn Schmidt, einen Bäckermeister aus Wuppertal. Er ist vor einigen Wochen auf dem Rückweg von der Arbeit mit dem Auto verunglückt und hat dadurch ein Bein verloren. Nach einer zweiwöchigen stationären Behandlung ist nun eine komplizierte Therapie mit anschließender beruflicher Neuorientierung geplant.

Nachdem er die Nachricht zur Kenntnis genommen hat, will Herr Müller einen neuen Fall für das Monitoring der Therapie starten. Hierzu öffnet er das adCM-Werkzeug. Direkt zu Beginn muss er sich authentifizieren, damit er später nur seine eigenen Annotationen sehen kann. Anschließend sieht er ein Dialogfenster mit einer Übersicht über alle noch nicht abgeschlossenen Fälle sowie zwei Optionen zur Erstellung eines neuen Falls. Sie stellen Herrn Müller vor die Wahl, den Fall mit einer gänzlich neuen Agenda, oder mit einem Template zu starten.

8.4.3 *T₁: START DES FALLS*

Herr Müller entscheidet sich dafür, die Arbeit mit einem Template zu beginnen und wählt aus einer Liste das Template mit den Grundschritten aus Abbildung 75 aus. Das Werkzeug lädt die Agenda mit allen fünf Einträgen und zeigt den dazu passenden Workspace in der Treemap-Ansicht an (siehe Abbildung 77).

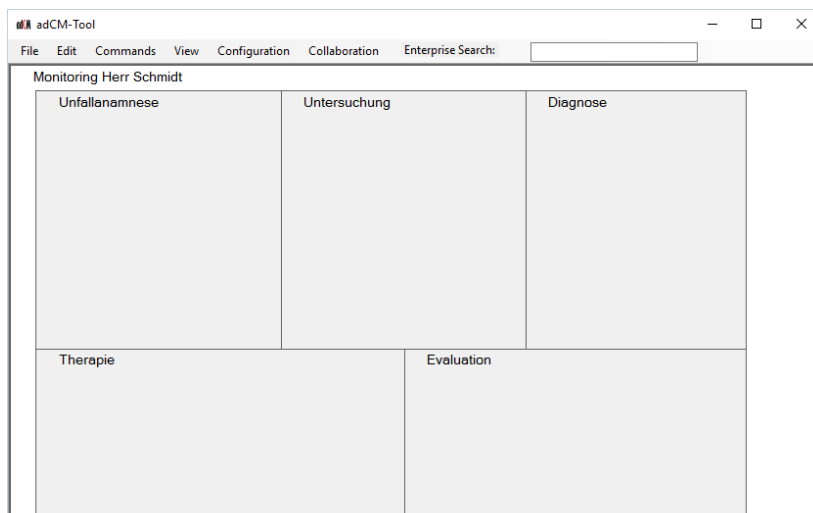


Abbildung 77: Zustand des Workspace nach dem Start des neuen Falls

Mit dieser Aktion erzeugt das System im Hintergrund automatisch eine Reihe von Ereignissen und legt die Ereignisse in der Protokolldatei ab. Das erste Ereignis dokumentiert den Start des Falls (siehe Listing 11). Dessen Attribute geben später Aufschluss darüber, von welchem Element des adCM-Metamodells das Ereignis stammt (*sourcetype*), um welche konkrete Objektinstanz es dabei geht (*sourceid*), welche Art von Ereignis stattgefunden hat (*eventtype*) und zuletzt, zu welchem Zeitpunkt das Ereignis ausgelöst wurde. Ab dem zweiten Ereignis folgen fünf Ereignisse, die das automatische Befüllen der Agenda mit den Einträgen aus dem Template dokumentieren. Eines dieser Ereignisse ist exemplarisch ebenfalls im Listing 11 abgebildet.



Listing 11: Zustand des Ereignisprotokolls unmittelbar nach Start des Falls (Ausschnitt)

Zugleich werden die Ereignisse über die Message-Oriented Middleware an die Graphdatenbank weitergeleitet. Dort angekommen, wird anhand der Ereignisse eine Graphstruktur erzeugt, die den aktuellen Zustand des Falls mit seinen noch unerledigten Agendaeinträgen widerspiegelt.

Herr Müller beginnt die Bearbeitung des Falls zunächst damit, die vorliegenden Daten in die Agenda zu integrieren. Hierzu zählen unter anderem zwei Dokumente: ein Unfallbericht, den er in den Workspace des Agendaeintrags „Unfallanamnese“ ablegt sowie der Heilverlaufsplan, den er in den Workspace „Therapie“ ablegt. Außerdem trägt er in der Agenda unter „Diagnose“ den ICD-Code für die traumatische Amputation zwischen Knie und oberem Sprunggelenk ein („ICD S 88.1“).

8.4.4 *T₂: MODERNE THERAPIEMETHODEN RECHERCHIEREN*

Nachdem Herr Müller den aktuellen Sachstand in die Agenda einsortiert hat, kann er mit der Begleitung der Therapie beginnen. Aufgrund seiner Erfahrung und mit Blick auf seine Rolle als Reha-Leistungsträger möchte er die im Heilverlaufsplan dokumentierten Therapieschritte zunächst nachvollziehen, bevor er sie in die Agenda übernimmt. Plangemäß soll zuerst eine Therapie der noch verbleibenden Schmerzen am Beinstumpf und dann die Therapie der Phantomschmerzen erfolgen. Sobald die Prothese hergestellt und angepasst wurde, soll die Reha mit einer Gangschulung fortgesetzt und mit dem erfolgreichen Einüben von Alltagsbewegungen abgeschlossen werden. Weil Herr Müller selbst kein Arzt ist, ist er auf zusätzliche Informationen angewiesen, um diese Schritte zu verifizieren. Hierzu telefoniert er mit einem Gutachter und verwendet die Unternehmenssuchmaschine, die einerseits an die verschiedenen Datenquellen des Unternehmens angeschlossen ist, andererseits aber auch extern bereitgestellte Daten aus dem

Internet anbietet. Auf diesem Wege erhält er neben dem Feedback des Gutachters auch ausführliche Beschreibungen über moderne Therapiemaßnahmen. Sie stammen sowohl aus Webseiten von Selbsthilfvereinen als auch aus einer Publikation des europäischen Branchenverbandes der Prothesenhersteller, die auf dem unternehmensinternen Dokumentenserver liegt. Herr Müller legt diese Informationen als Artefakte in den passenden Workspace ein und verknüpft sie mit dem Heilverlaufsplan (siehe Abbildung 78).

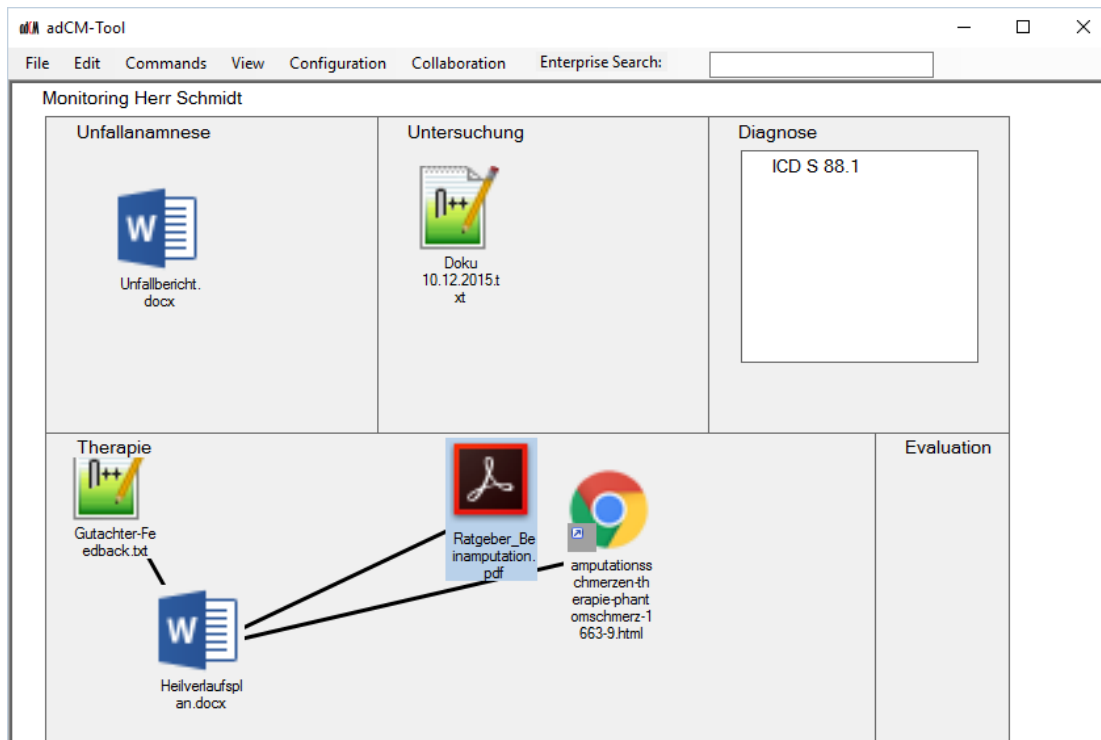


Abbildung 78: Strukturierung der Rechercheergebnisse zu den Therapiemaßnahmen

Mithilfe der Werkzeugunterstützung zur Erstellung von Annotationen kennzeichnet Herr Müller anschließend in der Publikation aufgezählte Leistungen, die möglicherweise auch für Herrn Schmidt infrage kommen. Dazu zählt die Optimierung der prothetischen Versorgung, zu der Herr Müller den Umfang und die Kosten mit der Vertragsklinik abklären will (siehe Abbildung 79, links). Auf einem Online-Portal für Amputationspatienten findet er zudem Informationen über die Spiegeltherapie zur Linderung von Phantomschmerzen. Auch hier macht er sich eine Notiz auf der Webseite mithilfe des adCM-Werkzeugs (siehe Abbildung 79, rechts).

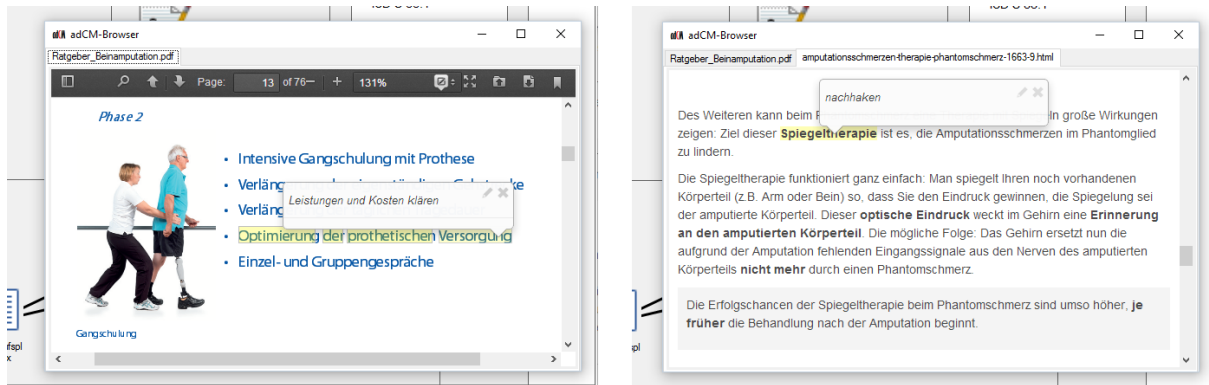


Abbildung 79: Reflektieren und Recherchieren von Therapiemaßnahmen mit einem adCM-Werkzeug

Herr Müller bespricht diese letzten noch offenen Punkte vor Therapiebeginn mit den behandelnden Ärzten, der Vertragsklinik sowie mit Herrn Schmidt selbst.

8.4.5 T₃: THERAPIEERFOLG REFLEKTIEREN

Während der Therapie erkundigt sich Herr Müller in regelmäßigen Abständen bei allen beteiligten nach dem Erfolg der Therapiemaßnahmen. Dabei führt er jeweils Einzelgespräche mit den Ärzten und dem Patienten. Die Ergebnisse dokumentiert er mithilfe von Annotationen im aktuellen Heilverlaufsplan. Dort nutzt er vor allem die Bewertungsfunktion, mit der er zusätzlich zum Kommentar auf einer Skala von 1-5 Sternen den Erfolg der Maßnahme einschätzt.

Die Therapie von Herrn Schmidt verläuft zunächst nach Plan. Nur gelegentlich kommt es während der Maßnahmen zu Veränderungen, die in enger Abstimmung mit den Ärzten und dem Patienten umgesetzt werden. Doch nur wenige Tage nach Beginn der Gehschule häufen sich Verhaltensauffälligkeiten von Herrn Schmidt. Er zeigt eine massive Übererregbarkeit während der Übungen, die eine Fortsetzung der Maßnahme verhindern. Zudem klagt er über Schlafstörungen. Die Klinik meldet diese Vorgänge an Herrn Müller, der umgehend einen lokalen Facharzt für psychosomatische Medizin hinzuzieht und die Agenda um den Eintrag „Psychologische Untersuchung“ ergänzt.

8.4.6 T₄: PSYCHISCHES KRANKHEITSBILD

Nach einer umfassenden Sitzung mit Herrn Schmidt diagnostiziert der Facharzt eine posttraumatische Belastungsstörung (PTBS), die wahrscheinlich durch den schweren Autounfall verursacht wurde. Das verzögerte Auftreten der Symptome sei laut Facharzt durchaus üblich. Als konkreter Auslöser kommt ein Sturz während der Gehschule infrage, der als Trigger des traumatischen Erlebnisses gewirkt hat. Der Arzt empfiehlt, die aktuellen Maßnahmen zugunsten einer psychologischen Therapie zu unterbrechen.

Aufgrund der neuen Diagnose notiert sich Herr Müller zunächst den entsprechenden ICD-Code („ICD F 43.1“) in der Agenda. In diesem Moment wechselt im Hintergrund der Recommender-Algorithmus den Zustand von Stufe 0 auf Stufe 1. Ab diesem Zeitpunkt schlägt der Recommender-Algorithmus beim Hinzufügen neuer Agendaeinträge mithilfe der Autovervollständigung Einträge vor, die Teil des Template sind. Denn mit dem zusätzlichen Agendaeintrag umfasst die Agenda acht Einträge und ist damit groß genug, um die Zahl der verfügbaren Templates für diese Funktion einzuschränken. Dieser Schwellwert ist vom Prozessverantwortlichen bei der Einrichtung des adCM-Werkzeugs festgelegt worden – ebenso wie der Schwellwert für die Übereinstimmungen zwischen der Agenda und einem beliebigen Template, bevor es vorgeschlagen werden darf (Stufe 2). Zum aktuellen Zeitpunkt gibt es zwischen der Agenda und dem zum Zeitpunkt t_n erstellten Template aber erst sechs Übereinstimmungen (siehe Abbildung 80). Deshalb verbleibt der Algorithmus zunächst in Stufe 1.

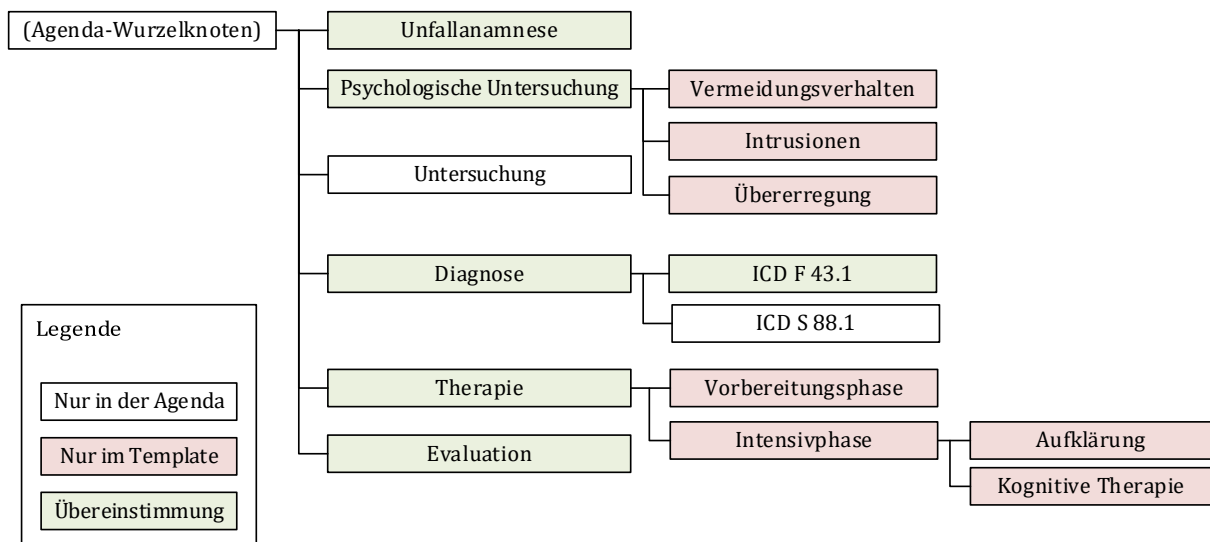


Abbildung 80: Abgleich zwischen der aktuellen Agenda und dem Template

Unterdessen ergänzt Herr Müller die Symptome aus der psychologischen Untersuchung. Hierzu fügt er einen neuen Eintrag „Schlafstörungen“ hinzu. Während der Eingabe kann der Recommender keine lexikalische Übereinstimmung mit einem der Templateelemente erkennen. Infolgedessen kann er kein Template-Element vorschlagen. Doch als Herr Müller damit beginnt, die Übererregung ebenfalls der Agenda hinzuzufügen, erkennt der Recommender die Übereinstimmung mit dem gleichnamigen Eintrag im Template und bietet folglich bereits nach dem ersten Buchstaben die Autovervollständigung des Wortes an. Herr Müller übernimmt den Vorschlag und fügt den Eintrag der Agenda hinzu. Dadurch erhöht sich die Anzahl der übereinstimmenden Einträge von sechs auf sieben. Damit ist der Schwellwert erreicht und der

Recommender-Algorithmus schlägt das gesamte Template vor. Weil es das einzige Template mit sieben Übereinstimmungen ist, werden keine anderen Templates zusätzlich angeboten.

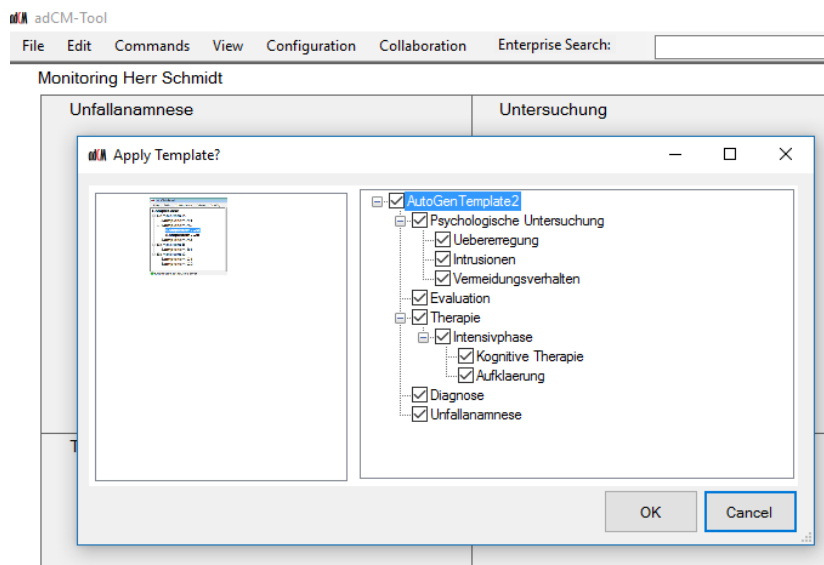


Abbildung 81: Dialog des Recommender-Algorithmus, um das Template vorzuschlagen

Herr Müller überprüft die Template-Elemente eingehend. Er kommt zum Schluss, dass alle Einträge sowohl für die zukünftige Planung der eingeschobenen psychologischen Therapie als auch für die Qualitätssicherung der bereits erfolgten psychologischen Untersuchung interessant erscheinen. Beispielsweise will er den Facharzt danach fragen, ob er den Patienten nicht nur hinsichtlich der Schlafstörungen und der Übererregung befragt beziehungsweise untersucht hat, sondern ob der Patient auch weitere Symptome wie Intrusionen oder Vermeidungsverhalten zeigt¹¹. Darüber hinaus erhält er mit der nun bereits vorliegenden Aufstellung der potentiellen Therapieschritte die Möglichkeit, den in Kürze erwarteten Therapieplan des Facharztes zu überprüfen.

8.4.7 T_M : THERAPIE ABGESCHLOSSEN

In Zusammenarbeit mit den Ärzten und dem Team in der Klinik ist es Herrn Müller gelungen, seinem Patienten eine Perspektive zu verschaffen, wie er mit seiner veränderten Lebenssituation zurechtkommen kann. Im Rahmen der psychologischen und der physiologischen Therapie wurden Herrn Schmidt Maßnahmen aufgezeigt, die ihm sowohl den Umgang mit seinen körperlichen Einschränkungen als auch mit seinem Trauma und dessen Symptomen erleichtern können.

¹¹ Intrusionen sind „unwillkürliche und belastende Erinnerungen an das Trauma“ Maercker 2013, S. 14.

Zum Abschluss der Therapie fand ein gemeinsames Gespräch mit dem Durchgangsarzt und dem Patienten statt, in dem der Erfolg der Therapie reflektiert und die weiteren Schritte zur beruflichen Umschulung grob abgesprochen wurden. Herr Müller legt das Protokoll dieses Gesprächs sowie einen ausführlichen Evaluationsbericht des Therapieverlaufs in den entsprechenden Workspace und markiert den Fall als geschlossen. Im letzten Schritt archiviert er den Fall. Mit dem dadurch ausgelösten Ereignis werden die protokollierten Informationen über den Fallverlauf und dessen Agendastruktur in die Fallbasis übertragen. Dort angelangt, werden sie vom Discovery-Algorithmus wieder zu Erzeugung neuer Templates herangezogen.

Die finale Agenda des abgeschlossenen Falls ist in Abbildung 82 dargestellt. Sie enthält 16 Einträge, wovon drei von Herrn Müller einzeln manuell hinzugefügt wurden. Die restlichen 13 Einträge stammen aus zwei importierten Templates: dem Basis-Template mit den grundlegenden Schritten der Therapiebeobachtung und dem Template mit besonderen Schritten bei einer PTBS. Von diesen 13 Einträgen sind 11 relevant gewesen, weil sie von Herrn Müller bearbeitet, beziehungsweise als erledigt markiert wurden. Die übrigen zwei Einträge (Vermeidungsverhalten und Intrusionen) sind hingegen nicht relevant gewesen. Dabei handelt es sich um zwei PTBS-Symptome, die zwar vom Template vorgeschlagen und von Herrn Müller für den Import in die Agenda ausgewählt wurden. Es hat sich aber im Verlauf des Falls herausgestellt, dass diese Symptome bei der Untersuchung keine Rolle gespielt haben, weil die festgestellte Übererregung und der Schlafmangel bereits für eine PTBS-Diagnose ausgereicht haben.

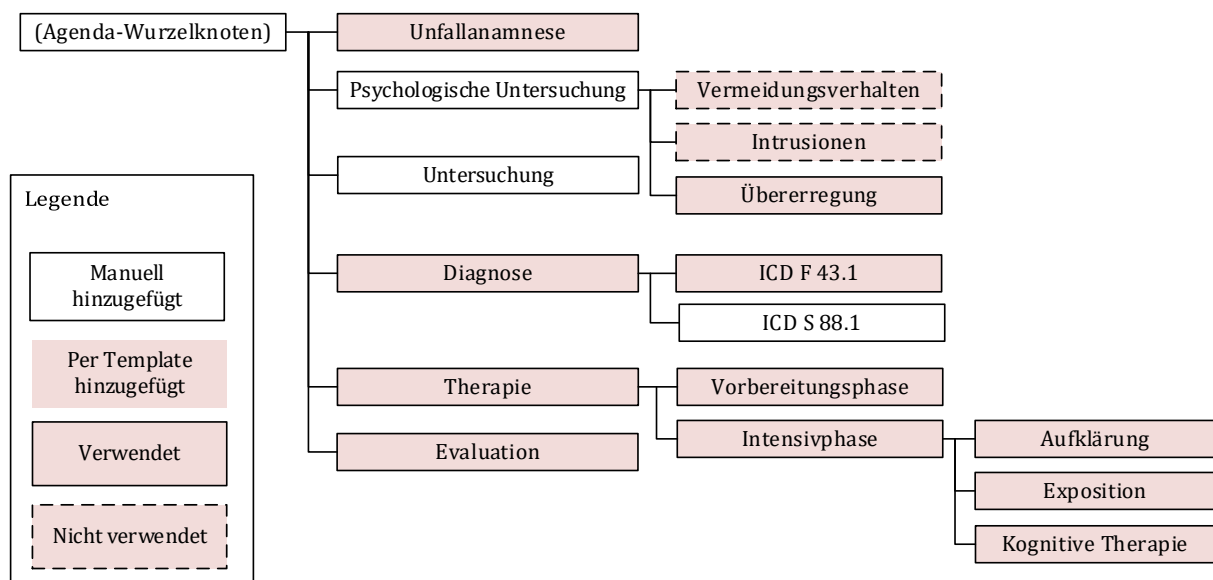


Abbildung 82: Finale Agenda des Fallbeispiels mit Angaben zur Herkunft und Verwendung

An diesem Punkt können die in Kapitel 6 beschriebenen Metriken als Indikatoren für die Qualität der Unterstützungsfunktion in diesem Beispiel herangezogen werden. Die Fitness beschreibt, wie

viele der verwendeten Agendaeinträge durch ein Template vorgeschlagen wurden. Zur Erinnerung: Sie errechnet hierzu das Verhältnis der Zahl der vorgeschlagenen (V) und verwendeten (B) Einträge zur Zahl der verwendeten Einträge einer Agenda insgesamt:

Formel 11: Formel zur Errechnung der Fitness Q_F und der Präzision Q_P

$$Q_F = \frac{|V \cap B|}{|B|}, \quad Q_P = \frac{|V \cap B|}{|V|}$$

Weil von 14 verwendeten Einträgen insgesamt 11 vorgeschlagen wurden, beträgt die Fitness knapp 0,79. Das heißt, fast 79% aller verwendeten Einträge in der Agenda wurden zuvor durch ein Template vorgeschlagen. Die Precision beschreibt, wie viele der vorgeschlagenen Templates tatsächlich relevant waren. Sie bemisst sich aus dem Verhältnis der Zahl der verwendeten und vorgeschlagenen Einträge zur Zahl der vorgeschlagenen Einträge insgesamt. Weil von insgesamt 13 vorgeschlagenen Einträgen 11 tatsächlich verwendet wurden, bemisst sich die Precision auf knapp 0,85. Das heißt, fast 85% aller vorgeschlagenen Einträge waren tatsächlich relevant.

8.5 ZUSAMMENFASSUNG UND BEWERTUNG

In diesem Kapitel wurde mithilfe eines einfachen Beispiels deutlich gemacht, wie die einzelnen Kernaktivitäten von Case Managern durch Konzepte und Methoden des Agenda-driven Case Managements unterstützt werden können. Dabei konnten zwei Aspekte veranschaulicht werden: Erstens wurde verdeutlicht, wie mithilfe der adCM-Elemente Agenda, Workspace und Artefakt die Organisation des Falls im Allgemeinen und das Zusammentragen, Strukturieren und Bewerten von unterschiedlichen Informationen in einer zentralen Fallakte im Speziellen unterstützt werden. Zweitens wurde ein Beispiel gezeigt, wie etablierte Planungsmuster in Form von Templates erzeugt werden können und unter welchen Bedingungen sie in einem konkreten Fall dazu beitragen, den Case Manager in der Planung seines Falls zu unterstützen.

Es wurde am Beispiel dargelegt, wie die Agenda als eine Datenstruktur eingesetzt werden kann, um den Case Manager dazu zu befähigen, den Verlauf des Falls zu planen und entlang dieses Plans alle benötigten Informationen den einzelnen Phasen zuzuordnen. Ebenfalls wurde veranschaulicht, inwiefern sie dem Case Manager die benötigte Flexibilität bietet, um auf unvorhergesehene Ereignisse wie eine plötzlich eintretende posttraumatische Belastungsstörung reagieren und den Prozess nach eigenem Ermessen neu strukturieren zu können.

Es wurde ebenfalls herausgestellt, wie die beiden eng verbundenen Konzepte des Workspaces und der Artefakte von den technischen Details der Daten (Format, Ablageort, Zugriffsrechte)

abstrahieren und an einem konkreten Beispiel den Case Manager dazu befähigen, eine universelle Fallakte mit artefaktübergreifend einheitlichen Funktionen zu pflegen. So war es Herrn Müller möglich, die unterschiedlichen Informationen über den Patienten und über den Therapieverlauf zentral abzulegen und bei Bedarf miteinander in Beziehung zu setzen, zu kommentieren und zu bewerten.

Die Templates selbst und die Art und Weise wie sie in den Fall eingebracht werden, haben sich in dem Beispiel als ein Konzept erwiesen, das den Case Manager nicht nur bei der Ausarbeitung nachfolgender Handlungsschritte unterstützt. Das Template diente im konkreten Beispiel auch als eine Art Checkliste, mit der der Case Manager trotz fehlender medizinischer Ausbildung den Therapieplan des Arztes überprüfen konnte.

Das Beispiel hat aber auch deutlich gemacht, dass die umgesetzten Konzepte zur Unterstützung von Case Managern nur ein Ausschnitt dessen sind, was mit den protokollierten Daten über den Fallverlauf an Unterstützung erzielt werden kann. Denn der Discovery-Algorithmus berücksichtigte beispielsweise nur die statische Struktur der Agenda nach Abschluss eines Falls. Hätte Herr Müller beispielsweise im Fallverlauf einen Agendaeintrag hinzugefügt und vor Abschluss des Falls wieder gelöscht, würde dieser Eintrag in zukünftigen Anwendungen des Mining-Algorithmus ignoriert werden. Die Analyse solcher Veränderungen einer Agenda sind aber dank der im Monitoring-Konzept vorgesehenen semantisch verknüpften Protokollierung möglich und bilden einen interessanten Anknüpfungspunkt für weiterführende Arbeiten.

Abschließend muss darauf hingewiesen werden, dass es sich sowohl bei dem beschriebenen Beispiel als auch bei den Fällen in der Fallbasis nicht um real existierende Fälle handelt. Die Ergebnisse lassen daher keine unmittelbaren und zwingenden Rückschlüsse auf andere Beispiele zu. Sie erlauben lediglich Aussagen über die Umsetzbarkeit der adCM-Konzepte und geben Anhaltspunkte dafür, unter welchen Bedingungen die Konzepte einen Case Manager unterstützen. So lässt sich beispielsweise festhalten, dass der Mining-Algorithmus nur dann zu einem Ergebnis kommt, wenn es Gemeinsamkeiten in den einzelnen Agenden gibt. Sind die Fälle so unterschiedlich, dass es keine identischen Teilbäume gibt, die die Anforderungen an deren Größe und Häufigkeit erfüllen, werden keine Templates erzeugt beziehungsweise vorgeschlagen.

9 FAZIT UND AUSBLICK

In diesem Kapitel werden die einzelnen Beiträge der Dissertation zusammengefasst. Es wird jeweils diskutiert, inwiefern die in der Zielstellung unter Kapitel 4.1 aufgeführten Punkte durch das Agenda-driven Case Management erreicht werden. Zum Schluss werden mögliche Anknüpfungspunkte für weiterführende Arbeiten aufgezeigt.

9.1 FAZIT

9.1.1 ADCM-METAMODELL

Das adCM-Metamodell bildet die konzeptionelle Grundlage für die Lösungsbausteine. Die darin enthaltenen Konzepte prägen jeweils einen bestimmten Grundgedanken des adCM aus und adressieren bestimmte Tätigkeiten eines Case Managers. In diesem Sinne stellt die **Agenda** die grundlegende Datenstruktur dar, mit der ein Case Manager die Aktivitäten in einem Fall so flexibel strukturieren kann, wie es der individuelle Kontext des Falls erfordert. Der Case Manager ist damit in der Lage, den gedanklich vorstrukturierten Plan zu externalisieren und weiterzuentwickeln. Auf diese Weise trägt das adCM-Metamodell zur Erreichung des ersten Ziels der Dissertation bei, nämlich der Unterstützung von Planungsaktivitäten in der Wissensarbeit (siehe 4.1.1).

Der **Workspace** und die darin enthaltenen **Artefakte** setzen diesen Grundgedanken der Flexibilität fort und übertragen ihn auf die digitale Fallakte. Mit diesen beiden Konzepten kann der Case Manager auf eine Ablage zurückgreifen, die ihm eine individuelle Organisation der recherchierten Inhalte unabhängig von ihrer konkreten Beschaffenheit oder Herkunft bietet. Dabei können nicht nur die Zusammenhänge zwischen den Artefakten abgebildet werden, sondern auch zwischen den Artefakten und der Agenda: Weil ein Workspace immer zu einem Agendaeintrag gehört, hat der Case Manager einen Überblick darüber, welche Artefakte zur Erreichung der jeweiligen (Teil-)Ziele eines Falls bearbeitet oder berücksichtigt werden müssen. Dadurch wird die Grundlage für die Erreichung eines weiteren Ziels der Dissertation erreicht, nämlich der Unterstützung des Archivierens und Strukturierens von Inhalten aus heterogenen Datenquellen in einer gemeinsamen Fallakte (siehe 4.1.3).

Mit der Einführung des **Template**-Begriffs wurde eine Datenstruktur geschaffen, die der Definition von Erfahrungswissen über einen Prozess dient. Weil ihr grundlegender Aufbau der Agenda gleicht, verfügt sie über eine ähnliche Flexibilität: Templates können sowohl Muster für allgemeine, grundlegende Schritte oder Ziele eines Prozesses enthalten als auch sehr spezielle Verfahrensweisen, die nur in einem konkreten Kontext relevant sind. Im Gegensatz zu imperativen Prozessmodellen sind sie weder verbindlich, noch geben sie eine feste Reihenfolge für die Abarbeitung der Schritte vor. Werden Templates im Prozessverlauf vorgeschlagen, unterstützen sie den Case Manager bei der Fortentwicklung seines Plans (siehe 4.1.1).

9.1.2 MONITORING

Die Kernaufgabe dieses Bausteins ist das Protokollieren von Benutzerinteraktionen in einem wissensintensiven Prozess mit dem Ziel, eine ausreichend große Datenbasis für das anschließende Discovery zu schaffen. Mit dem vorgestellten Lösungskonzept ist es gelungen, diese Informationen standardkonform und anwendungsübergreifend einheitlich zu protokollieren. Dabei wurde eine Abstraktionsebene gewählt, die im Gegensatz zu vergleichbaren Ansätzen keine zusätzlichen Verarbeitungsschritte erfordern, zum Beispiel das Konsolidieren zusammengehöriger, eher technischer Ereignisse zu einer übergeordneten fachlichen Aktivität. Zudem ist es gelungen, die Ereignisse mit einer Semantik zu versehen, indem sie in einer graphbasierten Datenbank mit den zugehörigen Konzepten im adCM-Metamodell verknüpft wurden. Diese Design-Entscheidung bietet zudem die Grundlage dafür, beliebige Konzepte (zum Beispiel domänenspezifische Artefakttypen) zu ergänzen und mit den Ereignisdaten automatisch zu verknüpfen. Auf diese Weise sind die Daten offen für präzisere Abfragen, die über Erkenntnisse über die allgemeinen Konzepte (zum Beispiel häufig verwendete Agendaeinträge oder Artefakte) hinausgehen. Am Beispiel des Reha-Managements kann etwa untersucht werden, ob bei jedem Abschluss einer Therapiemaßnahme ein Gesprächsprotokoll angefertigt wurde. Die Implementierung dieses Aspekts ist allerdings nicht mehr Bestandteil dieser Dissertation und wird im Ausblick aufgegriffen.

Der Monitoring-Lösungsbaustein bildet mit den Ereignisprotokoll-Komplexitätsmetriken zugleich die Schnittstelle zwischen dem adCM-Ansatz und dem etablierten Process Mining-Workflow. Denn mit den Metriken bietet es eine Hilfestellung dafür erkennen zu können, ob ein Prozess im Sinne Loeffelers Klassifikation (Loeffeler et al. 1998) strukturiert genug ist, um durch die etablierten Methoden des Process Mining-Ansatzes analysiert werden zu können.

Die Evaluation des Bausteins hat ergeben, dass die in der Zielstellung aufgeführten Anforderungen bis auf kleinere Abweichungen umgesetzt werden konnten. Anhand des

Fallbeispiels aus Kapitel 8 konnte außerdem gezeigt werden, dass die protokollierten Daten von dem Discovery-Algorithmus ausgelesen und so verarbeitet werden konnten, dass Muster in der Fallausführung identifiziert werden konnten. In diesem Sinne liefert der Monitoring-Baustein die Grundlagen für eine proaktive Unterstützung des Case Managers bei der Planung und Organisation seiner Aktivitäten (siehe 4.1.1).

9.1.3 DISCOVERY

Die wesentliche Aufgabe des Discovery-Lösungsbausteins ist die Gewinnung von Informationen über den ausgeführten Prozess auf Grundlage der aus dem Monitoring bezogenen Daten. Unter Zuhilfenahme von Technologien des Semantic Web wurde ein mehrschichtiges Datenmodell entwickelt, das die Ereignisdaten sowohl mit den allgemeinen Konzepten aus dem adCM-Metamodell verknüpfen kann als auch domänenspezifische Erweiterungen dieser Semantik erlaubt. Auf diese Weise können die Ereignisdaten Aufschluss darüber geben, wie sich ein Fall und insbesondere dessen Agenda über den Verlauf der Zeit entwickelt hat. Wird ein Fall abgeschlossen – erkennbar am entsprechenden Ereignis –, wird sein finaler Zustand (Agenda, Artefakte, ...) in die Menge der abgeschlossenen Fälle übertragen. Dort angelangt, wird er zusammen mit den anderen Fällen vom adCM Miner auf häufig wiederkehrende Muster in der Prozessausführung untersucht. Diese Muster werden als Templates ebenfalls in dem Datenmodell abgelegt.

Die Evaluation des adCM Miners erfolgte in einem Experiment und sollte in zwei Schritten zunächst die grundsätzliche Funktionstüchtigkeit und dann die Nützlichkeit des Algorithmus evaluieren. Hierzu wurde ein Ereignisprotokoll mit 1143 Fällen ausgewählt, das in der Process Mining-Disziplin als Datensatz für Leistungsvergleiche zwischen Discovery-Algorithmen bereitgestellt wurde. Um die Ergebnisse mit dem Stand der Technik zu vergleichen, wurden die Daten sowohl mit dem adCM Miner analysiert als auch mit ausgewählten Algorithmen aus dem Process Mining.

Zur Feststellung der Funktionstüchtigkeit des adCM Miners wurde untersucht, ob der Algorithmus dazu dienen kann, Muster in Form von Templates in den Daten zu erkennen – unabhängig von der Nützlichkeit dieser Muster bei der Unterstützung von Planungsaktivitäten. Hierzu wurde eine Hälfte des Datensatzes im Sinne der Kreuzvalidierung als Trainingsdatensatz entnommen (ca. 570 Fälle). Weil der Algorithmus unabhängig von den gewählten Parametern stets eine Vielzahl an Templates erzeugte, wurde das erste Ziel der Evaluation erreicht. Schon in diesem ersten Schritt der Evaluation scheiterten einige der ausgewählten Process Mining-Algorithmen an der Komplexität des Protokolls. Sie waren ohne vorangestellte Verfahren zur

Filterung seltener Aktivitäten oder Pfade nicht in der Lage, ein zusammenhängendes Prozessmodell zu erzeugen.

Um die Nützlichkeit der gewonnenen Templates zu evaluieren, wurden die ca. 570 übrig gebliebenen Fälle in Form einer Simulation nachgestellt (Replay). Gab es währenddessen eine teilweise Übereinstimmung zwischen der jeweiligen Agenda und mindestens einem Template, wurden sie vorgeschlagen und mithilfe eines sehr einfachen Algorithmus ausgewählt und in die Agenda integriert. Im Verlauf der Simulation wurde pro Fall festgehalten, wie viele Agendaeinträge ursprünglich durch ein Template vorgeschlagen wurden und wie viele davon wiederum tatsächlich im weiteren Verlauf verwendet wurden. Auf diese Weise konnte die Nützlichkeit der Templates und damit auch die Nützlichkeit des adCM Miners errechnet werden. Je nach Konfiguration des Algorithmus wurden 60-70% der vorgeschlagenen Agenda-Einträge vorhergesagt und waren in diesem Sinne „nützlich“. Verglichen mit den in diesem Evaluationsschritt noch verbliebenen Process Mining-Algorithmen ist das ein sehr gutes Ergebnis. Allerdings geht diese Präzision zulasten der Fitness, deren Wert bei den Process Mining-Algorithmen durchweg höher liegt. Eine hohe Fitness kann erzielt werden, wenn dem Wissensarbeiter sehr häufig sehr viele Vorschläge unterbreitet werden. Geht man allerdings davon aus, dass jeder Vorschlag den Wissensarbeiter in seiner Arbeit unterbricht, ist die Fitness im Kontext der Wissensarbeit der Präzision unterzuordnen.

Das Ergebnis der Evaluation untermauert, dass die in diesem Baustein vorgestellten Konzepte zur Gewinnung von Templates dazu beitragen können, den Case Manager bei der Identifikation zukünftiger Schritte zur Bearbeitung des Falls zu unterstützen. Damit wurde das erste Ziel der Dissertation erreicht.

9.1.4 IMPLEMENTIERUNG

Die oben beschriebenen Lösungsbausteine wurden durch ein zentrales Werkzeug für Case Manager implementiert. Hierzu wurde eine Architektur entwickelt, die neben dem vorrangigen Ziel, den Case Manager bei der strukturierten Externalisierung und Fortentwicklung seines Plans mithilfe der Agenda und den Templates zu unterstützen, auch weitere Ziele verfolgt. Hierzu zählt beispielsweise die Unterstützung von Recherchetätigkeiten eines Case Managers durch die Vereinfachung des Datenzugriffs auf vielfältige Datenquellen. Die konzeptionelle Grundlage hierfür bildet der Artefaktbegriff. Er verbirgt gegenüber dem Case Manager Details über die konkreten technischen Eigenschaften des Artefakts, beispielsweise ob es sich um eine PDF-Datei, eine Notiz oder eine Webseite handelt. Darauf aufbauend wird in der softwaretechnischen

Architektur des Werkzeugs ein zentraler Service integriert, der die vielen möglichen Datenquellen mithilfe einer Unternehmenssuchmaschine und einem Konnektor-Framework zugreifbar macht.

Auch die Erarbeitung von Konzepten zur Unterstützung des Case Managers beim Archivieren und Strukturieren zuvor recherchierter Daten ist ein Ziel dieser Dissertation. Hierzu dient der Workspacebegriffs als konzeptionelle Basis, die im Rahmen der Werkzeugimplementierung mithilfe eines Benutzerschnittstellenkonzepts ergänzt und konkretisiert wurde. Im Ergebnis wurde eine besondere Form einer digitalen Fallakte konzipiert und umgesetzt, auf die der Case Manager die Artefakte miteinander in Beziehung setzen kann. Unter Verwendung einer Treemap-Darstellung behält der Case Manager dabei stets den Überblick über alle Artefakte eines Falls.

Ein weiteres Ziel bestand darin, den Case Manager beim Reflektieren der (Zwischen-)Ergebnisse oder Handlungsschritte zu unterstützen. Hierzu wurde das Konzept der Annotationen entwickelt, mit denen der Case Manager Agendaeinträge und Artefakte mit Notizen versehen kann. Die softwaretechnische Herausforderung lag hierbei darin, dass der Artefaktbegriff von technischen Details wie dem Dateiformat oder Zugriffsrechten abstrahiert. Die technische Umsetzung der Annotationen musste also so ausgearbeitet werden, dass beispielsweise auch Webseiten oder schreibgeschützte Office-Dokumente annotiert werden können. Hierfür wurde eine Lösung erarbeitet, nach der Annotationen mithilfe einer anwendungsspezifischen Adapter-Architektur erstellt und auf einen zentralen Annotations-Service abgelegt werden.

Um den adCM-Lebenszyklus vom Monitoring über das Discovery bis zur operativen Unterstützung des Case Managers zu schließen, wurde zudem ein 3-stufiger Algorithmus zum Vorschlagen von Templates beschrieben und implementiert. Dessen Verhalten in den verschiedenen Stufen richtet sich an der Gewissheit aus, inwiefern eine identifizierte Menge mutmaßlich kontextrelevanter Templates tatsächlich relevant ist. Hierzu werden die Templateelemente zunächst nur im Rahmen einer unaufdringlichen Autovervollständigungsfunktion empfohlen und erst dann vollumfänglich vorgeschlagen, wenn diese Autovervollständigung vom Case Manager angenommen wird. Im Rahmen der Unterstützung des Case Managers bei der Fortentwicklung des Plans trägt dieses Konzept dazu bei, den Zeitpunkt der Vorschläge so zu wählen, dass der Case Manager durch die Vorschläge nicht in seiner Arbeit unterbrochen wird.

9.2 AUSBLICK

Aufgrund der notwendigen inhaltlichen Fokussierung der Dissertation gibt es einige Themenbereiche, in denen der vorgestellte Ansatz weiter ausgebaut werden kann. Sie werden in den nachfolgenden Abschnitten einzeln erläutert.

9.2.1 DURCHFÜHRUNG VERSCHIEDENER FALLSTUDIEN

Wie zu Beginn des achten Kapitels deutlich gemacht wurde, sind die adCM-Konzepte bisher nur in Form einzelner technischer Simulationen sowie eines übergreifenden Fallbeispiels untersucht worden. Die Ergebnisse aus den Simulationen geben zwar Aufschluss über die Nützlichkeit ausgewählter Algorithmen in einer Umgebung mit überwiegend kontrollierbaren Eigenschaften, sie erlauben aber keine Aussage über die Nützlichkeit für Wissensarbeiter im Praxiseinsatz. Auch das mithilfe des Werkzeugs durchgeführte Fallbeispiel testete nur die grundsätzliche Machbarkeit des adCM-Ansatzes an einem speziell ausgewählten Szenario.

Um die Nützlichkeit in der Praxis nachvollziehbar darzulegen, bedarf es darüber hinaus eine langfristige Anwendung der adCM-Konzepte in der Praxis in Form von mehreren unabhängig durchgeführten Fallstudien. Sie müssen untersuchen, in welchem Maße die entwickelten Funktionen die Wissensarbeiter im Kontext realer Fälle unterstützen. Hierzu muss die prototypische Implementierung des Werkzeugs zu einer Anwendung weiterentwickelt werden, die über die reine Funktionalität hinaus Qualitätseigenschaften erfüllt, die im Produktivbetrieb wichtig sind. Zum Beispiel müssen Eigenschaften wie die Zuverlässigkeit des Werkzeugs verbessert werden, damit die Wissensarbeiter während der Arbeit nicht mit schlecht nachvollziehbaren Fehlermeldungen konfrontiert werden. Auch die Benutzbarkeit der grafischen Oberfläche muss an den aktuellen Stand der Technik angepasst werden, um eine einfachere oder schneller erlernbare Bedienung zu erzielen.

9.2.2 MONITORING: WEITERE METRIKEN FÜR EREIGNISPROTOKOLLE

In dieser Dissertation wurden Metriken vorgestellt, um die Komplexität des Case Managements anhand der Ereignisprotokolle feststellen zu können. Doch nach (Ukelson 2010) gibt es weitere Eigenschaften solcher Prozesse, die jeweils ebenfalls Gegenstand einer oder mehrerer Metriken sein könnten. Demnach besteht das Case Management beispielsweise hauptsächlich aus menschlicher Interaktion (Kollaboration, Verhandlung). Um solche Interaktionsmuster feststellen zu können, müssen die Attribute der Ereignisse untersucht werden, die Details über die ausführende Person oder Organisationseinheit beschreiben. Weiterhin hat laut Ukelson jeder

Fall seinen eigenen Case Manager. Auch dieser Zusammenhang ließe sich mit einer eigenen Metrik untersuchen.

9.2.3 *DISCOVERY: WEITERE ANALYSEZIELE*

Der in dieser Dissertation vorgestellte Algorithmus zur Erzeugung von Templates ist nur ein Beispiel, wie die Informationen aus dem Monitoring verarbeitet werden können. Beispielsweise berücksichtigt der adCM Miner nicht die Reihenfolge, in der die Agendaeinträge abgearbeitet werden. Das graphbasierte Datenmodell (siehe Kapitel 6.3.2) erlaubt aber eine Analyse dieser Informationen, weil zu jedem Fall nicht bloß die finale Agenda selbst, sondern auch alle Ereignisse gespeichert werden, die die zeitliche Entwicklung der Agenda beschreiben. Weiterführende Arbeiten können den adCM Miner so anpassen, dass er den Case Manager darauf hinweist, welche der vorgeschlagenen Templateelemente üblicherweise zuerst durchgeführt werden. Hierzu müsste der zugrundeliegende Frequent Subtree Mining-Algorithmus durch eine Alternative ausgetauscht werden, die die Ordnung der Knoten berücksichtigt.

Darüber hinaus bieten die Ereignisdaten nicht nur Informationen über die Agenda, sondern auch über die Artefakte. Diese Informationen kann ein Algorithmus beispielsweise nutzen, um wichtige Input-Artefakte zu finden. Hierzu kann er nach Artefakten suchen, die häufig geöffnet und durchsucht, aber nicht manipuliert beziehungsweise gespeichert wurden. Umgekehrt kann er auch nach häufig manipulierten und gespeicherten Artefakten suchen, um wichtige Output-Artefakte zu erkennen. Solche Zusatzinformationen über die Artefakte können den Case Manager zur Laufzeit unterstützen. Beim Starten eines Falls könnten beispielsweise kurz die wichtigsten Input-Artefakte angezeigt werden, während beim Schließen des Falls mit einer Liste der häufigsten Output-Artefakte abgesichert werden kann, ob nicht ein wesentliches Artefakt vergessen wurde. Prozessverantwortliche können diese Informationen ebenfalls verwenden, um herauszufinden, ob bestimmte Artefakte Teil einer verbindlichen Prozessdokumentation werden sollten.

Eine Weiterführung der hier vorgestellten Konzepte kann auch darin bestehen zu untersuchen, ob ein neu hinzugefügtes Artefakt in der Vergangenheit oft mit einem anderen Artefakt in Beziehung stand. Sofern dieses „verwandte“ Artefakt noch nicht Bestandteil des betroffenen Workspace ist, kann ein Algorithmus diesen Zusammenhang erkennen und mithilfe des adCM-Werkzeugs darauf hinweisen. Dadurch kann das Strukturieren von Artefakten mit zusätzlichem Prozesswissen unterstützt werden.

Wie in Kapitel 2.2.2.4 beschrieben, gehört zu den Aktivitäten des Reflektierens auch das Bewerten von Artefakten. Falls also Artefakte in der Vergangenheit besonders gut beziehungsweise besonders schlecht bewertet wurden, kann dies jeweils ein Indiz dafür sein, dass sie besonders wertvoll beziehungsweise ungeeignet für den Prozess sind. Diese Informationen können beim Template Mining berücksichtigt werden, um bestimmte Artefakte bei der Zusammenstellung der Templates ein- oder auszuschließen.

9.2.4 MENSCHENLESBARE MODELLIERUNG SCHWACH-STRUKTURIERTER PROZESSE

Die Process Mining-Community hat es erreicht, rund um das Werkzeug ProM eine Art Ökosystem zu etablieren, in dem Forscher aus aller Welt Plug-Ins für ihre eigenen Mining-Algorithmen anbinden können. Diese Plug-Ins lesen ein standardkonformes Ereignisprotokoll ein und geben ein Modell aus. Dabei kann das Protokoll zwar durchaus komplex beziehungsweise unstrukturiert sein. Wie in dieser Dissertation gezeigt wurde, sind Process Mining-Algorithmen aber nur bedingt in der Lage, ein menschenlesbares Prozessmodell aus diesem Protokoll zu erzeugen. Doch auch der adCM Miner bietet mit den vielen Templates keine Möglichkeit, den Prozess übersichtlicher darzustellen. Es fehlt also ein Mining-Algorithmus, dessen Ergebnis ein menschenlesbares Modell eines schwach-strukturierten Prozesses ist.

In Ergänzung zu dem hier vorgestellten Lösungsansatz könnte sich eine weiterführende Arbeit also damit befassen, das oben beschriebene Ökosystem von Plug-Ins um einen Algorithmus zu erweitern, dessen Ergebnis ein CMMN-Modell ist. Nach (Kurz et al. 2015) könnte dieser Algorithmus beispielsweise die temporal-logischen Zusammenhänge in dem Protokoll analysieren. Demnach kann auch die Visualisierung des Fortschritts eines Falls trotz fehlendem strikten Kontrollfluss durch einen CMMN-basierten Replay-Algorithmus ermöglicht werden.

9.2.5 CONFORMANCE: DEFINITION UND EINHALTUNG VON GESCHÄFTSREGELN

Bei dem Design und der Implementierung des mehrschichtigen Datenmodells (Kapitel 6.3.2) wurde eine graphbasierte Form der Speicherung gewählt. Als Motivation für diese Entscheidung wurde angeführt, dass mit dem Linked Data-Ansatz die darin beschriebenen Modelle und Metamodelle einfach erweitert werden können. Beispielsweise können die adCM-Konzepte um domänenspezifische Konzepte ergänzt werden.


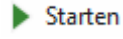

Zusätzlich zu dieser Motivation wurde diese Form der Datenhaltung aber auch deshalb gewählt, damit Prozessverantwortliche Geschäftsregeln für das Case Management definieren und Auswertungen über die Einhaltung dieser Regeln mithilfe von Inferenzmaschinen durchführen können. Die Geschäftsregeln werden in einer formalen Sprache wie RDFS oder OWL beschrieben

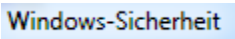
und beziehen sich dabei auf vorher definierte (domänenspezifische) Konzepte, beispielsweise einen Agendaeintrag oder einen Heilverlaufsplan. Auf diese Weise können Zielzustände eines Falls definiert werden, die in jedem Fall vor Abschluss erreicht werden müssen. Um Prozessverantwortliche die Einarbeitung in diese Sprachen zu ersparen, wurde in (Benner-Wickner et al. 2015b) ein Prototyp für eine Benutzerschnittstelle vorgestellt. Zukünftige Arbeiten können an diese Ergebnisse anknüpfen und damit auf die Unterstützung von Prozessverantwortlichen im Case Management abzielen.

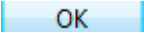
10 ANHANG




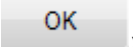
10.1 SIKULI-TESTFALL

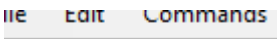
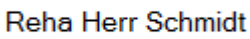
```


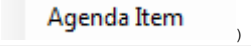

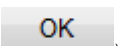
# start
sleep(1)

click(
)
click()
click()
wait(
, 60)

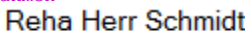

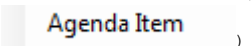

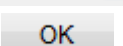
# Login

wait(
, 15)

click()

# new Case

click(
)

type(
, "Reha Herr Schmidt")

type(
, "Reha Management")

click(
)

# hvp
wait(1)



rightClick(
)

click(
)

click(
)

type(
, "Heilverlaufsplan")

click(
)

# Röntgenaufnahmen

rightClick(
)

click(
)

click(
)

type(
, "Roentgenaufnahmen")

click(
)

# Artefakte hinzufügen

```

```

rightClick(Pattern(Heilverlaufsplan).targetOffset(300,50))
click(Import)
click(File)
click(copy here)
click(trunk)
click(AnnServer
Automated User Acceptance Tests
CefSharp)
doubleClick(test artifacts)
doubleClick(Diagnose.txt)

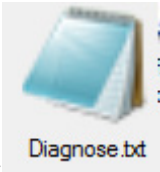
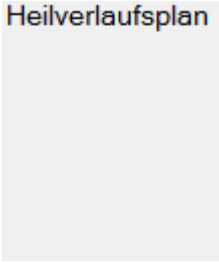
rightClick(Pattern(Heilverlaufsplan).similar(0.80).targetOffset(200,200))
click(Import)
click(File)
click(copy here)
click(trunk)
click(AnnServer
Automated User Acceptance Tests
CefSharp)
doubleClick(test artifacts)
doubleClick(Medikation.txt)

rightClick(Pattern(Heilverlaufsplan).targetOffset(100,50))
click(Import)
click(File)
click(copy here)
click(trunk)
click(AnnServer
Automated User Acceptance Tests
CefSharp)
doubleClick(test artifacts)
doubleClick(Heilverlaufsplan.docx)

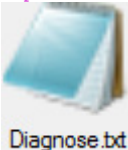
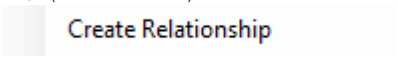


# move


```

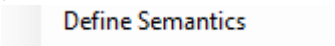
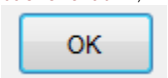
```

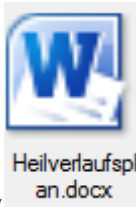
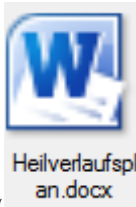
dragDrop(

,

)

wait(1)

# relationship

rightClick(

)
click(

)
click(

)

rightClick(

)

wait(1)
click(

)
type("basiert auf")
click(

)

# open

doubleClick(

)

wait(4)


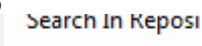
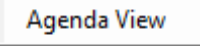

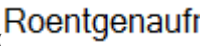
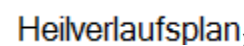
#rightClick("1437303255997.png")
#click("1437303282995.png")

type(Key.F4, Key.ALT)

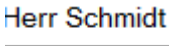
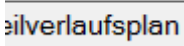
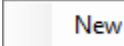
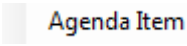



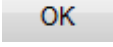
```

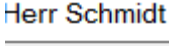
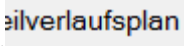
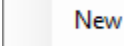


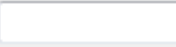

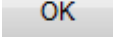
```

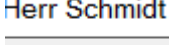
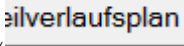
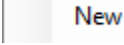
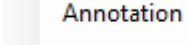
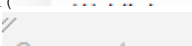
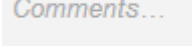
wait(2)

# reorganize
#click("1437247629085.png")

click(


click(Pattern(

).similar(0.71))
dragDrop(

,

)

type(Key.ALT + Key.RIGHT + Key.RIGHT + Key.RIGHT + Key.ENTER + Key.DOWN + Key.ENTER)

# berufliche Reha


rightClick(

)
click(

)
click(

Name:

)
type(

, "berufliche Reha")
click(

)

# ambulante Physiotherapie


rightClick(

)
click(

)
click(

Name:

)
type(

, "ambulante Physiotherapie")
click(

)

# annotation


rightClick(

)
click(

)
click(

)
wait(

, 10)

type("hier weitermachen" + Key.TAB + "Notiz" + Key.ENTER)

```

```
# pause

#type(Key.ALT)

#type(Key.ALT)

#wait(2)

#type(Key.ENTER + Key.DOWN + Key.DOWN + Key.DOWN + Key.ENTER)

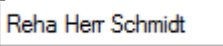
wait(5)

type("w", Key.CTRL)
```

```
# re-open

wait(1)

type("o", Key.CTRL)

doubleClick()
```

```
# find



doubleClick()



wait()

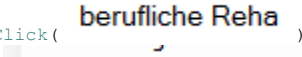
type("f", Key.CTRL)

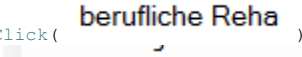
type("Diagnose" + Key.ENTER)

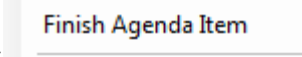
type(Key.F4, Key.ALT)

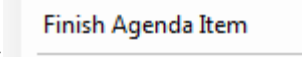
type(Key.F4, Key.ALT)
```

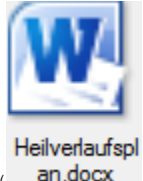
```
# manipulate artifact

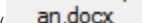


rightClick()



click()



doubleClick()

wait(5)

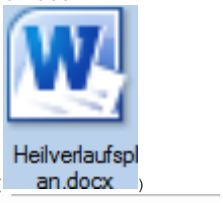
type("Planaenderung")

type("s", Key.CTRL)

type(Key.F4, Key.ALT)
```



```
# rename artifact
```



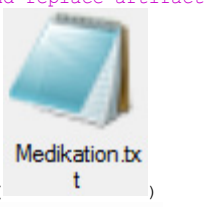
```
rightClick(
click(
type("Heilverlaufspl_2015.07.19")
type(Key.ENTER)
```

```
# annotate artifact
```



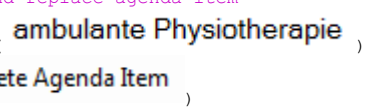
```
rightClick(
click(
wait(
type("Problem mit Medikament X" + Key.TAB + "Komplikation" + Key.ENTER)
```

```
# delete and replace artifact
```

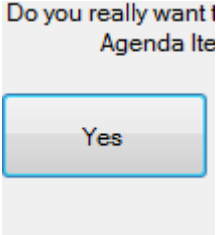


```
rightClick(
click(
rightClick(
click(
click(
wait(1)
type("Medikation" + Key.ENTER)
```

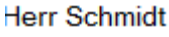
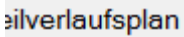

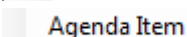

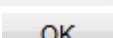
```
# delete and replace agenda item
```

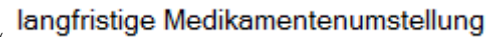
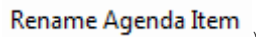
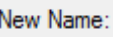


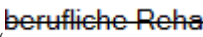
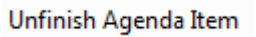
```
rightClick(
click(
```

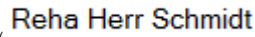
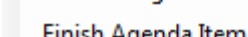


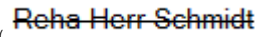
```

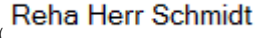
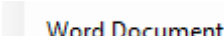
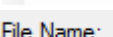
click()
rightClick()
click()
click()
type(, "langfristige Medikamentenumstellung")
click()

# rename agenda item
rightClick()
click()
wait()
type("Medikamentenumstellung" + Key.ENTER)

# unfinish agenda item
rightClick()
click()

# finish agenda
rightClick()
click()

# unfinish agenda
rightClick()
click()

# add final report
rightClick()
click()
click()
wait()

```

```
type("Abschlussbericht" + Key.ENTER)

# finish agenda and archive case
Reha Herr Schmidt
rightClick(
    Finish Agenda Item
)
click(
    File
)
click(
    Archive current Case
)
```

10.2 EREIGNISPROTOKOLL

```
<log xes.version="1.4" xes.features="nested-attributes" openxes.version="1.8"
xmlns="http://www.xes-standard.org/">
  <extension name="Concept" prefix="concept" uri="http://www.xes-standard.org/concept.xesext" />
</>
  <extension name="Time" prefix="time" uri="http://www.xes-standard.org/time.xesext" />
  <extension name="ID" prefix="identity" uri="http://www.xes-standard.org/identity.xesext" />
  <extension name="OPC" prefix="opc" uri="http://www.paluno.de/uni-due/opc.xesext" />
  <global scope="trace">
    <string key="opc:caseid" value="string" />
  </global>
  <global scope="event">
    <string key="opc:eventid" value="string" />
    <string key="opc:sourcetype" value="string" />
    <string key="opc:sourceid" value="string" />
    <string key="opc:eventtype" value="string" />
    <string key="opc:userid" value="string" />
    <string key="opc:clientid" value="string" />
  </global>
  <trace>
    <string key="opc:caseid" value="620c9847-a6a4-44f3-b335-91f8f513dd12" />
    <event xmlns="">
      <string key="opc:caseid" value="620c9847-a6a4-44f3-b335-91f8f513dd12" />
      <date key="time:timestamp" value="2015-10-21T18:39:47.14" />
      <id key="opc:eventid" value="ce0f5912-5ac6-460b-924a-8ce703ae66b4" />
      <string key="opc:sourcetype" value="case" />
      <string key="opc:sourceid" value="Reha Herr Schmidt" />
      <string key="opc:eventtype" value="started" />
      <string key="opc:businessProcessType" value="Reha Management" />
      <id key="opc:userid" value="1000000001" />
      <id key="opc:clientid" value="MILANKODOROOF2A" />
      <boolean key="opc:systemEvent" value="false" />
    </event>
  </trace>
  <trace>
    <event xmlns="">
      <string key="opc:caseid" value="620c9847-a6a4-44f3-b335-91f8f513dd12" />
      <date key="time:timestamp" value="2015-10-21T18:39:47.14" />
      <id key="opc:eventid" value="6996368c-37c3-4150-b1c5-1cc21d6892db" />
      <string key="opc:sourcetype" value="agenda" />
      <string key="opc:sourceid" value="Reha Herr Schmidt" />
      <string key="opc:eventtype" value="started" />
      <id key="opc:userid" value="1000000001" />
      <id key="opc:clientid" value="MILANKODOROOF2A" />
      <boolean key="opc:systemEvent" value="false" />
    </event>
    <event>
      <string key="opc:caseid" value="620c9847-a6a4-44f3-b335-91f8f513dd12" />
      <date key="time:timestamp" value="2015-10-21T18:39:54.23" />
      <id key="opc:eventid" value="3a6bf41d-ce44-431a-88ec-6599f30832b5" />
      <string key="opc:sourcetype" value="item" />
      <string key="opc:sourceid" value="Heilverlaufsplan" />
      <string key="opc:eventtype" value="added" />
      <string key="opc:parent" value="Reha Herr Schmidt/" />
      <id key="opc:userid" value="1000000001" />
    </event>
  </trace>
</log>
```

```

        <id key="opc:clientid" value="MILANKODOROF2A" />
        <boolean key="opc:systemEvent" value="false" />
    </event>
    <event>
        <string key="opc:caseid" value="620c9847-a6a4-44f3-b335-91f8f513dd12" />
        <date key="time:timestamp" value="2015-10-21T18:40:00.66" />
        <id key="opc:eventid" value="85be2c8e-d460-496d-a61a-bf3787df1853" />
        <string key="opc:sourcetype" value="item" />
        <string key="opc:sourceid" value="Roentgenaufnahmen" />
        <string key="opc:eventtype" value="added" />
        <string key="opc:parent" value="Reha Herr Schmidt/" />
        <id key="opc:userid" value="1000000001" />
        <id key="opc:clientid" value="MILANKODOROF2A" />
        <boolean key="opc:systemEvent" value="false" />
    </event>
    <event>
        <string key="opc:caseid" value="620c9847-a6a4-44f3-b335-91f8f513dd12" />
        <date key="time:timestamp" value="2015-10-21T18:40:09.86" />
        <id key="opc:eventid" value="b44947c1-54be-4105-b01b-2335aa92a7f0" />
        <string key="opc:sourcetype" value="artifact" />
        <string key="opc:sourceid" value="Diagnose.txt" />
        <string key="opc:eventtype" value="added" />
        <string key="opc:workspace" value="Reha Herr Schmidt/Heilverlaufsplan/" />
        <string key="uri" value="file:///C:/Users/Milanko/Desktop/Arbeit/OPC%20Toolset/trunk/OPCTool/bin/Debug/LocalStorage/Business%20Processes/Reha%20Management/Reha%20Herr%20Schmidt/Reha%20Herr%20Schmidt/Diagnose.txt" />
        <boolean key="opc:isLocal" value="True" />
        <string key="opc:artifactType" value="Document" />
        <id key="opc:userid" value="1000000001" />
        <id key="opc:clientid" value="MILANKODOROF2A" />
        <boolean key="opc:systemEvent" value="false" />
    </event>
    <event>
        <string key="opc:caseid" value="620c9847-a6a4-44f3-b335-91f8f513dd12" />
        <date key="time:timestamp" value="2015-10-21T18:40:17.94" />
        <id key="opc:eventid" value="a0d4ee98-elb4-4495-8124-66ccc00f1238" />
        <string key="opc:sourcetype" value="artifact" />
        <string key="opc:sourceid" value="Medikation.txt" />
        <string key="opc:eventtype" value="added" />
        <string key="opc:workspace" value="Reha Herr Schmidt/Heilverlaufsplan/" />
        <string key="uri" value="file:///C:/Users/Milanko/Desktop/Arbeit/OPC%20Toolset/trunk/OPCTool/bin/Debug/LocalStorage/Business%20Processes/Reha%20Management/Reha%20Herr%20Schmidt/Reha%20Herr%20Schmidt/Medikation.txt" />
        <boolean key="opc:isLocal" value="True" />
        <string key="opc:artifactType" value="Document" />
        <id key="opc:userid" value="1000000001" />
        <id key="opc:clientid" value="MILANKODOROF2A" />
        <boolean key="opc:systemEvent" value="false" />
    </event>
    <event>
        <string key="opc:caseid" value="620c9847-a6a4-44f3-b335-91f8f513dd12" />
        <date key="time:timestamp" value="2015-10-21T18:40:26.20" />
        <id key="opc:eventid" value="9f55494b-dc2d-4a0d-a5cf-85cf74770dd8" />
        <string key="opc:sourcetype" value="artifact" />
        <string key="opc:sourceid" value="Heilverlaufsplan.docx" />
        <string key="opc:eventtype" value="added" />
        <string key="opc:workspace" value="Reha Herr Schmidt/Heilverlaufsplan/" />
        <string key="uri" value="file:///C:/Users/Milanko/Desktop/Arbeit/OPC%20Toolset/trunk/OPCTool/bin/Debug/LocalStorage/Business%20Processes/Reha%20Management/Reha%20Herr%20Schmidt/Reha%20Herr%20Schmidt/Heilverlaufsplan.docx" />
        <boolean key="opc:isLocal" value="True" />
        <string key="opc:artifactType" value="Document" />
        <id key="opc:userid" value="1000000001" />
        <id key="opc:clientid" value="MILANKODOROF2A" />
        <boolean key="opc:systemEvent" value="false" />
    </event>
    <event>
        <string key="opc:caseid" value="620c9847-a6a4-44f3-b335-91f8f513dd12" />
        <date key="time:timestamp" value="2015-10-21T18:40:28.30" />
        <id key="opc:eventid" value="2fc9292b-517b-4146-8843-99c2bfc707eb" />
        <string key="opc:sourcetype" value="item" />
        <string key="opc:sourceid" value="Heilverlaufsplan" />
        <string key="opc:eventtype" value="selected" />
    </event>

```

```

    <string key="opc:parent" value="Reha Herr Schmidt/" />
    <id key="opc:userid" value="1000000001" />
    <id key="opc:clientid" value="MILANKODOROOF2A" />
    <boolean key="opc:systemEvent" value="false" />
  </event>
  <event>
    <string key="opc:caseid" value="620c9847-a6a4-44f3-b335-91f8f513dd12" />
    <date key="time:timestamp" value="2015-10-21T18:40:31.75" />
    <id key="opc:eventid" value="b92d0699-945f-45eb-8b5c-2979559ab468" />
    <string key="opc:sourcetype" value="relationship" />
    <string key="opc:sourceid" value="Heilverlaufsplan.docx,Diagnose.txt" />
    <string key="opc:eventtype" value="added" />
    <string key="opc:src" value="Heilverlaufsplan.docx" />
    <string key="opc:dest" value="Diagnose.txt" />
    <string key="opc:workspace" value="Reha Herr Schmidt/Heilverlaufsplan/" />
    <id key="opc:userid" value="1000000001" />
    <id key="opc:clientid" value="MILANKODOROOF2A" />
    <boolean key="opc:systemEvent" value="false" />
  </event>
  <event>
    <string key="opc:caseid" value="620c9847-a6a4-44f3-b335-91f8f513dd12" />
    <date key="time:timestamp" value="2015-10-21T18:40:37.44" />
    <id key="opc:eventid" value="8de0aa78-9874-44e3-ac1d-f821dddafb63" />
    <string key="opc:sourcetype" value="artifact" />
    <string key="opc:sourceid" value="Heilverlaufsplan.docx" />
    <string key="opc:eventtype" value="opened" />
    <string key="opc:artifact" value="Heilverlaufsplan.docx" />
    <string key="uri" value="file:///C:/Users/Milanko/Desktop/Arbeit/OPC%20Toolset/trunk/OPCTool/bin/Debug/LocalStorage/Business%20Processes/Reha%20Management/Reha%20Herr%20Schmidt/Reha%20Herr%20Schmidt/Heilverlaufsplan.docx" />
    <string key="opc:applicationtype" value="WINWORD" />
    <string key="opc:workspace" value="Reha Herr Schmidt/Heilverlaufsplan/" />
    <id key="opc:userid" value="1000000001" />
    <id key="opc:clientid" value="MILANKODOROOF2A" />
    <boolean key="opc:systemEvent" value="false" />
  </event>
  <event>
    <string key="opc:caseid" value="620c9847-a6a4-44f3-b335-91f8f513dd12" />
    <date key="time:timestamp" value="2015-10-21T18:40:45.51" />
    <id key="opc:eventid" value="e7517c77-eb96-48c2-b487-fe4376e1419d" />
    <string key="opc:sourcetype" value="item" />
    <string key="opc:sourceid" value="Heilverlaufsplan" />
    <string key="opc:eventtype" value="selected" />
    <string key="opc:parent" value="Reha Herr Schmidt/" />
    <id key="opc:userid" value="1000000001" />
    <id key="opc:clientid" value="MILANKODOROOF2A" />
    <boolean key="opc:systemEvent" value="false" />
  </event>
  <event>
    <string key="opc:caseid" value="620c9847-a6a4-44f3-b335-91f8f513dd12" />
    <date key="time:timestamp" value="2015-10-21T18:40:47.34" />
    <id key="opc:eventid" value="96fa8d49-f3ec-47ec-8880-cd507baff575" />
    <string key="opc:sourcetype" value="item" />
    <string key="opc:sourceid" value="Roentgenaufnahmen" />
    <string key="opc:eventtype" value="selected" />
    <string key="opc:parent" value="Reha Herr Schmidt/" />
    <id key="opc:userid" value="1000000001" />
    <id key="opc:clientid" value="MILANKODOROOF2A" />
    <boolean key="opc:systemEvent" value="false" />
  </event>
  <event>
    <string key="opc:caseid" value="620c9847-a6a4-44f3-b335-91f8f513dd12" />
    <date key="time:timestamp" value="2015-10-21T18:40:47.41" />
    <id key="opc:eventid" value="03809c7f-0a2f-4f5e-8d40-6d909c37084d" />
    <string key="opc:sourcetype" value="item" />
    <string key="opc:sourceid" value="Heilverlaufsplan" />
    <string key="opc:eventtype" value="selected" />
    <string key="opc:parent" value="Reha Herr Schmidt/" />
    <id key="opc:userid" value="1000000001" />
    <id key="opc:clientid" value="MILANKODOROOF2A" />
    <boolean key="opc:systemEvent" value="false" />
  </event>
  <event>
    <string key="opc:caseid" value="620c9847-a6a4-44f3-b335-91f8f513dd12" />
    <date key="time:timestamp" value="2015-10-21T18:40:48.19" />

```

```

    <id key="opc:eventid" value="a569289f-8c38-479c-b1bd-32b1503a85b4" />
    <string key="opc:sourcetype" value="agenda" />
    <string key="opc:sourceid" value="Reha Herr Schmidt" />
    <string key="opc:eventtype" value="selected" />
    <string key="opc:parent" value="" />
    <id key="opc:userid" value="1000000001" />
    <id key="opc:clientid" value="MILANKODOROOF2A" />
    <boolean key="opc:systemEvent" value="false" />
  </event>
  <event>
    <string key="opc:caseid" value="620c9847-a6a4-44f3-b335-91f8f513dd12" />
    <date key="time:timestamp" value="2015-10-21T18:40:48.25" />
    <id key="opc:eventid" value="23997ce0-8497-49d5-8ad0-7a27e359e13f" />
    <string key="opc:sourcetype" value="item" />
    <string key="opc:sourceid" value="Roentgenaufnahmen" />
    <string key="opc:eventtype" value="moved" />
    <string key="opc:parent" value="Reha Herr Schmidt" />
    <string key="opc:targetParent" value="Heilverlaufsplan" />
    <id key="opc:userid" value="1000000001" />
    <id key="opc:clientid" value="MILANKODOROOF2A" />
    <boolean key="opc:systemEvent" value="false" />
  </event>
  <event>
    <string key="opc:caseid" value="620c9847-a6a4-44f3-b335-91f8f513dd12" />
    <date key="time:timestamp" value="2015-10-21T18:40:48.26" />
    <id key="opc:eventid" value="762e6ebe-b0c6-48c7-89f3-f078f8e76924" />
    <string key="opc:sourcetype" value="agenda" />
    <string key="opc:sourceid" value="Reha Herr Schmidt" />
    <string key="opc:eventtype" value="restructured" />
    <id key="opc:userid" value="1000000001" />
    <id key="opc:clientid" value="MILANKODOROOF2A" />
    <boolean key="opc:systemEvent" value="true" />
  </event>
  <event>
    <string key="opc:caseid" value="620c9847-a6a4-44f3-b335-91f8f513dd12" />
    <date key="time:timestamp" value="2015-10-21T18:40:49.52" />
    <id key="opc:eventid" value="826c01c7-b7ce-4dc9-b825-9931302b0ecd" />
    <string key="opc:sourcetype" value="item" />
    <string key="opc:sourceid" value="Heilverlaufsplan" />
    <string key="opc:eventtype" value="selected" />
    <string key="opc:parent" value="Reha Herr Schmidt/" />
    <id key="opc:userid" value="1000000001" />
    <id key="opc:clientid" value="MILANKODOROOF2A" />
    <boolean key="opc:systemEvent" value="false" />
  </event>
  <event>
    <string key="opc:caseid" value="620c9847-a6a4-44f3-b335-91f8f513dd12" />
    <date key="time:timestamp" value="2015-10-21T18:40:55.50" />
    <id key="opc:eventid" value="ad9ecb15-e210-44a3-b222-6619137ed5e8" />
    <string key="opc:sourcetype" value="item" />
    <string key="opc:sourceid" value="berufliche Reha" />
    <string key="opc:eventtype" value="added" />
    <string key="opc:parent" value="Reha Herr Schmidt/Heilverlaufsplan/" />
    <id key="opc:userid" value="1000000001" />
    <id key="opc:clientid" value="MILANKODOROOF2A" />
    <boolean key="opc:systemEvent" value="false" />
  </event>
  <event>
    <string key="opc:caseid" value="620c9847-a6a4-44f3-b335-91f8f513dd12" />
    <date key="time:timestamp" value="2015-10-21T18:41:02.92" />
    <id key="opc:eventid" value="2f938900-1740-42ff-b3c8-22aca68c48aa" />
    <string key="opc:sourcetype" value="item" />
    <string key="opc:sourceid" value="ambulante Physiotherapie" />
    <string key="opc:eventtype" value="added" />
    <string key="opc:parent" value="Reha Herr Schmidt/Heilverlaufsplan/" />
    <id key="opc:userid" value="1000000001" />
    <id key="opc:clientid" value="MILANKODOROOF2A" />
    <boolean key="opc:systemEvent" value="false" />
  </event>
  <event>
    <string key="opc:caseid" value="620c9847-a6a4-44f3-b335-91f8f513dd12" />
    <date key="time:timestamp" value="2015-10-21T18:41:11.11" />
    <id key="opc:eventid" value="66a5106a-be6b-4e56-886b-f667193418d5" />
    <string key="opc:sourcetype" value="item" />
    <string key="opc:sourceid" value="Heilverlaufsplan" />
    <string key="opc:eventtype" value="annotated" />
  </event>

```

```

        <id key="opc:userid" value="1000000001" />
        <id key="opc:clientid" value="MILANKODOROOF2A" />
        <boolean key="opc:systemEvent" value="false" />
    </event>
    <event>
        <string key="opc:caseid" value="620c9847-a6a4-44f3-b335-91f8f513dd12" />
        <date key="time:timestamp" value="2015-10-21T18:41:13.12" />
        <id key="opc:eventid" value="5084cbce-af3e-44c5-84b8-be340c9357ef" />
        <string key="opc:sourcetype" value="item" />
        <string key="opc:sourceid" value="Roentgenaufnahmen" />
        <string key="opc:eventtype" value="selected" />
        <string key="opc:parent" value="Reha Herr Schmidt/Heilverlaufsplan/" />
        <id key="opc:userid" value="1000000001" />
        <id key="opc:clientid" value="MILANKODOROOF2A" />
        <boolean key="opc:systemEvent" value="false" />
    </event>
    <event>
        <string key="opc:caseid" value="620c9847-a6a4-44f3-b335-91f8f513dd12" />
        <date key="time:timestamp" value="2015-10-21T18:41:13.17" />
        <id key="opc:eventid" value="163e44ba-a37b-45d6-bee6-9b8c669e5f4b" />
        <string key="opc:sourcetype" value="item" />
        <string key="opc:sourceid" value="berufliche Reha" />
        <string key="opc:eventtype" value="selected" />
        <string key="opc:parent" value="Reha Herr Schmidt/Heilverlaufsplan/" />
        <id key="opc:userid" value="1000000001" />
        <id key="opc:clientid" value="MILANKODOROOF2A" />
        <boolean key="opc:systemEvent" value="false" />
    </event>
    <event>
        <string key="opc:caseid" value="620c9847-a6a4-44f3-b335-91f8f513dd12" />
        <date key="time:timestamp" value="2015-10-21T18:41:13.22" />
        <id key="opc:eventid" value="e31211b6-568b-43f9-b134-611e1b7eef4e" />
        <string key="opc:sourcetype" value="item" />
        <string key="opc:sourceid" value="ambulante Physiotherapie" />
        <string key="opc:eventtype" value="selected" />
        <string key="opc:parent" value="Reha Herr Schmidt/Heilverlaufsplan/" />
        <id key="opc:userid" value="1000000001" />
        <id key="opc:clientid" value="MILANKODOROOF2A" />
        <boolean key="opc:systemEvent" value="false" />
    </event>
    <event>
        <string key="opc:caseid" value="620c9847-a6a4-44f3-b335-91f8f513dd12" />
        <date key="time:timestamp" value="2015-10-21T18:41:18.34" />
        <id key="opc:eventid" value="9a551b0f-f39e-448a-87e1-5c1d44cfbc0f" />
        <string key="opc:sourcetype" value="agenda" />
        <string key="opc:sourceid" value="Reha Herr Schmidt" />
        <string key="opc:eventtype" value="paused" />
        <id key="opc:userid" value="1000000001" />
        <id key="opc:clientid" value="MILANKODOROOF2A" />
        <boolean key="opc:systemEvent" value="false" />
    </event>
</trace>
<trace>
    <event xmlns="">
        <string key="opc:caseid" value="620c9847-a6a4-44f3-b335-91f8f513dd12" />
        <date key="time:timestamp" value="2015-10-21T18:41:20.84" />
        <id key="opc:eventid" value="4caf2d1a-a9ef-4f5f-8d2b-13d2b56dec1b" />
        <string key="opc:sourcetype" value="agenda" />
        <string key="opc:sourceid" value="Reha Herr Schmidt" />
        <string key="opc:eventtype" value="continued" />
        <id key="opc:userid" value="1000000001" />
        <id key="opc:clientid" value="MILANKODOROOF2A" />
        <boolean key="opc:systemEvent" value="false" />
    </event>
    <event>
        <string key="opc:caseid" value="620c9847-a6a4-44f3-b335-91f8f513dd12" />
        <date key="time:timestamp" value="2015-10-21T18:41:20.98" />
        <id key="opc:eventid" value="71f85adb-4792-4d7c-b330-58be18ee48fc" />
        <string key="opc:sourcetype" value="agenda" />
        <string key="opc:sourceid" value="Reha Herr Schmidt" />
        <string key="opc:eventtype" value="selected" />
        <string key="opc:parent" value="" />
        <id key="opc:userid" value="1000000001" />
        <id key="opc:clientid" value="MILANKODOROOF2A" />
        <boolean key="opc:systemEvent" value="false" />
    </event>

```



```

<event>
  <string key="opc:caseid" value="620c9847-a6a4-44f3-b335-91f8f513dd12" />
  <date key="time:timestamp" value="2015-10-21T18:41:22.16" />
  <id key="opc:eventid" value="3ece0a9d-8305-45b3-a164-5c4209931a02" />
  <string key="opc:sourcetype" value="artifact" />
  <string key="opc:sourceid" value="Diagnose.txt" />
  <string key="opc:eventtype" value="opened" />
  <string key="opc:artifact" value="Diagnose.txt" />
  <string key="uri" value="file:///C:/Users/Milanko/Desktop/Arbeit/OPC%20Toolset/trunk/OPCTool/bin/Debug/localStorage/Business%20Processes/Reha%20Management/Reha%20Herr%20Schmidt/Reha%20Herr%20Schmidt/Diagnose.txt" />
  <string key="opc:applicationtype" value="NOTEPAD" />
  <string key="opc:workspace" value="Reha Herr Schmidt/" />
  <id key="opc:userid" value="1000000001" />
  <id key="opc:clientid" value="MILANKODOROOF2A" />
  <boolean key="opc:systemEvent" value="false" />
</event>
<event>
  <string key="opc:caseid" value="620c9847-a6a4-44f3-b335-91f8f513dd12" />
  <date key="time:timestamp" value="2015-10-21T18:41:22.77" />
  <id key="opc:eventid" value="a5add357-19ec-402a-88f1-79f2950c0bb9" />
  <string key="opc:sourcetype" value="artifact" />
  <string key="opc:sourceid" value="Diagnose.txt" />
  <string key="opc:eventtype" value="searched in" />
  <id key="opc:userid" value="1000000001" />
  <id key="opc:clientid" value="MILANKODOROOF2A" />
  <boolean key="opc:systemEvent" value="false" />
</event>
<event>
  <string key="opc:caseid" value="620c9847-a6a4-44f3-b335-91f8f513dd12" />
  <date key="time:timestamp" value="2015-10-21T18:41:24.42" />
  <id key="opc:eventid" value="bdeb6ea3-e1b0-40cd-8784-15f85d3986c6" />
  <string key="opc:sourcetype" value="item" />
  <string key="opc:sourceid" value="berufliche Reha" />
  <string key="opc:eventtype" value="selected" />
  <string key="opc:parent" value="Reha Herr Schmidt/Heilverlaufsplan/" />
  <id key="opc:userid" value="1000000001" />
  <id key="opc:clientid" value="MILANKODOROOF2A" />
  <boolean key="opc:systemEvent" value="false" />
</event>
<event>
  <string key="opc:caseid" value="620c9847-a6a4-44f3-b335-91f8f513dd12" />
  <date key="time:timestamp" value="2015-10-21T18:41:25.69" />
  <id key="opc:eventid" value="f0ba7b74-3db7-4093-ad8c-d2471a42a15b" />
  <string key="opc:sourcetype" value="item" />
  <string key="opc:sourceid" value="berufliche Reha" />
  <string key="opc:eventtype" value="finished" />
  <string key="opc:parent" value="Reha Herr Schmidt/Heilverlaufsplan/" />
  <id key="opc:userid" value="1000000001" />
  <id key="opc:clientid" value="MILANKODOROOF2A" />
  <boolean key="opc:systemEvent" value="false" />
</event>
<event>
  <string key="opc:caseid" value="620c9847-a6a4-44f3-b335-91f8f513dd12" />
  <date key="time:timestamp" value="2015-10-21T18:41:27.08" />
  <id key="opc:eventid" value="3b53f9ec-e150-4a9b-b660-3529554af001" />
  <string key="opc:sourcetype" value="item" />
  <string key="opc:sourceid" value="Heilverlaufsplan" />
  <string key="opc:eventtype" value="selected" />
  <string key="opc:parent" value="Reha Herr Schmidt/" />
  <id key="opc:userid" value="1000000001" />
  <id key="opc:clientid" value="MILANKODOROOF2A" />
  <boolean key="opc:systemEvent" value="false" />
</event>
<event>
  <string key="opc:caseid" value="620c9847-a6a4-44f3-b335-91f8f513dd12" />
  <date key="time:timestamp" value="2015-10-21T18:41:28.06" />
  <id key="opc:eventid" value="4aelfc45-984a-4887-912c-2aeelbaecc7" />
  <string key="opc:sourcetype" value="artifact" />
  <string key="opc:sourceid" value="Diagnose.txt" />
  <string key="opc:eventtype" value="reorganized" />
  <string key="opc:workspace" value="Reha Herr Schmidt/Heilverlaufsplan/" />
  <string key="uri" value="file:///C:/Users/Milanko/Desktop/Arbeit/OPC%20Toolset/trunk/OPCTool/bin/Debug/localStorage/Business%20Processes/Reha%20Management/Reha%20Herr%20Schmidt/Reha%20Herr%20Schmidt/Diagnose.txt" />

```



```

age/Business%20Processes/Reha%20Management/Reha%20Herr%20Schmidt/Reha%20Herr%20Schmidt/Diagnos
e.txt" />
  <string key="opc:artifactType" value="Document" />
  <id key="opc:userid" value="1000000001" />
  <id key="opc:clientid" value="MILANKODOROOF2A" />
  <boolean key="opc:systemEvent" value="false" />
</event>
<event>
  <string key="opc:caseid" value="620c9847-a6a4-44f3-b335-91f8f513dd12" />
  <date key="time:timestamp" value="2015-10-21T18:41:28.70" />
  <id key="opc:eventid" value="1a1d23ed-b439-468e-ba5f-2ae91ec302cc" />
  <string key="opc:sourcetype" value="artifact" />
  <string key="opc:sourceid" value="Heilverlaufsplan.docx" />
  <string key="opc:eventtype" value="opened" />
  <string key="opc:artifact" value="Heilverlaufsplan.docx" />
  <string key="uri"
value="file:///C:/Users/Milanko/Desktop/Arbeit/OPC%20Toolset/trunk/OPCTool/bin/Debug/localStor
age/Business%20Processes/Reha%20Management/Reha%20Herr%20Schmidt/Reha%20Herr%20Schmidt/Heilver
laufsplan.docx" />
  <string key="opc:applicationtype" value="WINWORD" />
  <string key="opc:workspace" value="Reha Herr Schmidt/Heilverlaufsplan/" />
  <id key="opc:userid" value="1000000001" />
  <id key="opc:clientid" value="MILANKODOROOF2A" />
  <boolean key="opc:systemEvent" value="false" />
</event>
<event>
  <string key="opc:caseid" value="620c9847-a6a4-44f3-b335-91f8f513dd12" />
  <date key="time:timestamp" value="2015-10-21T18:41:38.78" />
  <id key="opc:eventid" value="66580b19-fcdb-4408-ba07-0d3f7e23ecbe" />
  <string key="opc:sourcetype" value="artifact" />
  <string key="opc:sourceid" value="Heilverlaufsplan_2015.07.19.docx" />
  <string key="opc:eventtype" value="renamed" />
  <string key="opc:originalName" value="Heilverlaufsplan.docx" />
  <id key="opc:userid" value="1000000001" />
  <id key="opc:clientid" value="MILANKODOROOF2A" />
  <boolean key="opc:systemEvent" value="false" />
</event>
<event>
  <string key="opc:caseid" value="620c9847-a6a4-44f3-b335-91f8f513dd12" />
  <date key="time:timestamp" value="2015-10-21T18:41:44.14" />
  <id key="opc:eventid" value="33f1f131-6e4d-41ee-a5d5-be8177a4c3a3" />
  <string key="opc:sourcetype" value="artifact" />
  <string key="opc:sourceid" value="Medikation.txt" />
  <string key="opc:eventtype" value="annotated" />
  <id key="opc:userid" value="1000000001" />
  <id key="opc:clientid" value="MILANKODOROOF2A" />
  <boolean key="opc:systemEvent" value="false" />
</event>
<event>
  <string key="opc:caseid" value="620c9847-a6a4-44f3-b335-91f8f513dd12" />
  <date key="time:timestamp" value="2015-10-21T18:41:46.31" />
  <id key="opc:eventid" value="4327f306-226c-48c2-82f1-87c2d4c20642" />
  <string key="opc:sourcetype" value="artifact" />
  <string key="opc:sourceid" value="Medikation.txt" />
  <string key="opc:eventtype" value="deleted" />
  <string key="opc:workspace" value="Reha Herr Schmidt/Heilverlaufsplan/" />
  <string key="uri"
value="file:///C:/Users/Milanko/Desktop/Arbeit/OPC%20Toolset/trunk/OPCTool/bin/Debug/localStor
age/Business%20Processes/Reha%20Management/Reha%20Herr%20Schmidt/Reha%20Herr%20Schmidt/Medikati
on.txt" />
  <string key="opc:artifactType" value="Document" />
  <id key="opc:userid" value="1000000001" />
  <id key="opc:clientid" value="MILANKODOROOF2A" />
  <boolean key="opc:systemEvent" value="false" />
</event>
<event>
  <string key="opc:caseid" value="620c9847-a6a4-44f3-b335-91f8f513dd12" />
  <date key="time:timestamp" value="2015-10-21T18:41:50.86" />
  <id key="opc:eventid" value="90d649e1-8173-4256-bc73-26187309883e" />
  <string key="opc:sourcetype" value="artifact" />
  <string key="opc:sourceid" value="Medikation.txt" />
  <string key="opc:eventtype" value="created" />
  <string key="opc:workspace" value="Reha Herr Schmidt/Heilverlaufsplan/" />
  <string key="uri"
value="file:///C:/Users/Milanko/Desktop/Arbeit/OPC%20Toolset/trunk/OPCTool/bin/Debug/localStor

```

age/Business%20Processes/Reha%20Management/Reha%20Herr%20Schmidt/Reha%20Herr%20Schmidt/Medikation.txt" />

```

    <boolean key="opc:isLocal" value="True" />
    <string key="opc:artifactType" value="Document" />
    <id key="opc:userid" value="1000000001" />
    <id key="opc:clientid" value="MILANKODOROOF2A" />
    <boolean key="opc:systemEvent" value="false" />
  </event>
  <event>
    <string key="opc:caseid" value="620c9847-a6a4-44f3-b335-91f8f513dd12" />
    <date key="time:timestamp" value="2015-10-21T18:41:51.74" />
    <id key="opc:eventid" value="9707238b-ecae-43ef-a809-35dfe872c04e" />
    <string key="opc:sourcetype" value="item" />
    <string key="opc:sourceid" value="ambulante Physiotherapie" />
    <string key="opc:eventtype" value="selected" />
    <string key="opc:parent" value="Reha Herr Schmidt/Heilverlaufsplan/" />
    <id key="opc:userid" value="1000000001" />
    <id key="opc:clientid" value="MILANKODOROOF2A" />
    <boolean key="opc:systemEvent" value="false" />
  </event>
  <event>
    <string key="opc:caseid" value="620c9847-a6a4-44f3-b335-91f8f513dd12" />
    <date key="time:timestamp" value="2015-10-21T18:41:54.30" />
    <id key="opc:eventid" value="156801d0-661d-4ba9-8ca8-f96b5dae35a9" />
    <string key="opc:sourcetype" value="item" />
    <string key="opc:sourceid" value="Heilverlaufsplan" />
    <string key="opc:eventtype" value="selected" />
    <string key="opc:parent" value="Reha Herr Schmidt/" />
    <id key="opc:userid" value="1000000001" />
    <id key="opc:clientid" value="MILANKODOROOF2A" />
    <boolean key="opc:systemEvent" value="false" />
  </event>
  <event>
    <string key="opc:caseid" value="620c9847-a6a4-44f3-b335-91f8f513dd12" />
    <date key="time:timestamp" value="2015-10-21T18:41:54.34" />
    <id key="opc:eventid" value="0c7af8a1-2705-46fb-9cb7-00a6e0d8fe8d" />
    <string key="opc:sourcetype" value="item" />
    <string key="opc:sourceid" value="ambulante Physiotherapie" />
    <string key="opc:eventtype" value="deleted" />
    <string key="opc:parent" value="Reha Herr Schmidt/Heilverlaufsplan/" />
    <id key="opc:userid" value="1000000001" />
    <id key="opc:clientid" value="MILANKODOROOF2A" />
    <boolean key="opc:systemEvent" value="false" />
  </event>
  <event>
    <string key="opc:caseid" value="620c9847-a6a4-44f3-b335-91f8f513dd12" />
    <date key="time:timestamp" value="2015-10-21T18:42:02.56" />
    <id key="opc:eventid" value="c27f8417-8450-493c-9d93-f25bdd4b6c10" />
    <string key="opc:sourcetype" value="item" />
    <string key="opc:sourceid" value="langfristige Medikamentenumstellung" />
    <string key="opc:eventtype" value="added" />
    <string key="opc:parent" value="Reha Herr Schmidt/Heilverlaufsplan/" />
    <id key="opc:userid" value="1000000001" />
    <id key="opc:clientid" value="MILANKODOROOF2A" />
    <boolean key="opc:systemEvent" value="false" />
  </event>
  <event>
    <string key="opc:caseid" value="620c9847-a6a4-44f3-b335-91f8f513dd12" />
    <date key="time:timestamp" value="2015-10-21T18:42:04.14" />
    <id key="opc:eventid" value="0d06b8e6-4ac8-4a2a-8c4c-8dfffc7c107bf" />
    <string key="opc:sourcetype" value="item" />
    <string key="opc:sourceid" value="langfristige Medikamentenumstellung" />
    <string key="opc:eventtype" value="selected" />
    <string key="opc:parent" value="Reha Herr Schmidt/Heilverlaufsplan/" />
    <id key="opc:userid" value="1000000001" />
    <id key="opc:clientid" value="MILANKODOROOF2A" />
    <boolean key="opc:systemEvent" value="false" />
  </event>
  <event>
    <string key="opc:caseid" value="620c9847-a6a4-44f3-b335-91f8f513dd12" />
    <date key="time:timestamp" value="2015-10-21T18:42:06.97" />
    <id key="opc:eventid" value="987a99e0-425d-4f51-ac41-dfe4c210c2a5" />
    <string key="opc:sourcetype" value="item" />
    <string key="opc:sourceid" value="Medikamentenumstellung" />
    <string key="opc:eventtype" value="renamed" />
    <string key="opc:originalName" value="langfristige Medikamentenumstellung" />
  </event>

```

```

    <string key="opc:parent" value="Reha Herr Schmidt/Heilverlaufsplan/" />
    <id key="opc:userid" value="1000000001" />
    <id key="opc:clientid" value="MILANKODOROF2A" />
    <boolean key="opc:systemEvent" value="false" />
  </event>
  <event>
    <string key="opc:caseid" value="620c9847-a6a4-44f3-b335-91f8f513dd12" />
    <date key="time:timestamp" value="2015-10-21T18:42:07.83" />
    <id key="opc:eventid" value="b90cb9d6-cbee-44cb-a7e2-c9ded6dfeb41" />
    <string key="opc:sourcetype" value="item" />
    <string key="opc:sourceid" value="berufliche Reha" />
    <string key="opc:eventtype" value="selected" />
    <string key="opc:parent" value="Reha Herr Schmidt/Heilverlaufsplan/" />
    <id key="opc:userid" value="1000000001" />
    <id key="opc:clientid" value="MILANKODOROF2A" />
    <boolean key="opc:systemEvent" value="false" />
  </event>
  <event>
    <string key="opc:caseid" value="620c9847-a6a4-44f3-b335-91f8f513dd12" />
    <date key="time:timestamp" value="2015-10-21T18:42:09.14" />
    <id key="opc:eventid" value="259551de-9a25-4319-9bf1-2bab225a497e" />
    <string key="opc:sourcetype" value="item" />
    <string key="opc:sourceid" value="berufliche Reha" />
    <string key="opc:eventtype" value="unfinished" />
    <string key="opc:parent" value="Reha Herr Schmidt/Heilverlaufsplan/" />
    <id key="opc:userid" value="1000000001" />
    <id key="opc:clientid" value="MILANKODOROF2A" />
    <boolean key="opc:systemEvent" value="false" />
  </event>
  <event>
    <string key="opc:caseid" value="620c9847-a6a4-44f3-b335-91f8f513dd12" />
    <date key="time:timestamp" value="2015-10-21T18:42:09.91" />
    <id key="opc:eventid" value="a6180474-d4fd-489f-9a25-5594dfce19c2" />
    <string key="opc:sourcetype" value="agenda" />
    <string key="opc:sourceid" value="Reha Herr Schmidt" />
    <string key="opc:eventtype" value="selected" />
    <string key="opc:parent" value="" />
    <id key="opc:userid" value="1000000001" />
    <id key="opc:clientid" value="MILANKODOROF2A" />
    <boolean key="opc:systemEvent" value="false" />
  </event>
  <event>
    <string key="opc:caseid" value="620c9847-a6a4-44f3-b335-91f8f513dd12" />
    <date key="time:timestamp" value="2015-10-21T18:42:11.20" />
    <id key="opc:eventid" value="84bd899b-b1c4-4752-a466-6f8ff4295df8" />
    <string key="opc:sourcetype" value="agenda" />
    <string key="opc:sourceid" value="Reha Herr Schmidt" />
    <string key="opc:eventtype" value="finished" />
    <string key="opc:parent" value="" />
    <id key="opc:userid" value="1000000001" />
    <id key="opc:clientid" value="MILANKODOROF2A" />
    <boolean key="opc:systemEvent" value="false" />
  </event>
  <event>
    <string key="opc:caseid" value="620c9847-a6a4-44f3-b335-91f8f513dd12" />
    <date key="time:timestamp" value="2015-10-21T18:42:13.08" />
    <id key="opc:eventid" value="586ad4b5-fdb0-4922-bd70-6a35c2377192" />
    <string key="opc:sourcetype" value="agenda" />
    <string key="opc:sourceid" value="Reha Herr Schmidt" />
    <string key="opc:eventtype" value="unfinished" />
    <string key="opc:parent" value="" />
    <id key="opc:userid" value="1000000001" />
    <id key="opc:clientid" value="MILANKODOROF2A" />
    <boolean key="opc:systemEvent" value="false" />
  </event>
  <event>
    <string key="opc:caseid" value="620c9847-a6a4-44f3-b335-91f8f513dd12" />
    <date key="time:timestamp" value="2015-10-21T18:42:20.22" />
    <id key="opc:eventid" value="ddb901ab-c55b-4f91-b96a-b68272eb6d1c" />
    <string key="opc:sourcetype" value="artifact" />
    <string key="opc:sourceid" value="Abschlussbericht.docx" />
    <string key="opc:eventtype" value="created" />
    <string key="opc:workspace" value="Reha Herr Schmidt/" />
    <string
      value="file:///C:/Users/Milanko/Desktop/Arbeit/OPC%20Toolset/trunk/OPCTool/bin/Debug/localStor
      key="uri"

```

```

age/Business%20Processes/Reha%20Management/Reha%20Herr%20Schmidt/Reha%20Herr%20Schmidt/Abschlussbericht.docx" />
  <boolean key="opc:isLocal" value="True" />
  <string key="opc:artifactType" value="Document" />
  <id key="opc:userid" value="1000000001" />
  <id key="opc:clientid" value="MILANKODOROOF2A" />
  <boolean key="opc:systemEvent" value="false" />
</event>
<event>
  <string key="opc:caseid" value="620c9847-a6a4-44f3-b335-91f8f513dd12" />
  <date key="time:timestamp" value="2015-10-21T18:42:21.36" />
  <id key="opc:eventid" value="2e572bb6-bcd7-4c1d-9d25-f022e8376f44" />
  <string key="opc:sourcetype" value="agenda" />
  <string key="opc:sourceid" value="Reha Herr Schmidt" />
  <string key="opc:eventtype" value="finished" />
  <string key="opc:parent" value="" />
  <id key="opc:userid" value="1000000001" />
  <id key="opc:clientid" value="MILANKODOROOF2A" />
  <boolean key="opc:systemEvent" value="false" />
</event>
<event>
  <string key="opc:caseid" value="620c9847-a6a4-44f3-b335-91f8f513dd12" />
  <date key="time:timestamp" value="2015-10-21T18:42:22.97" />
  <id key="opc:eventid" value="5a7b0636-8cd1-4b7f-ac43-fccfab0769b0" />
  <string key="opc:sourcetype" value="agenda" />
  <string key="opc:sourceid" value="Reha Herr Schmidt" />
  <string key="opc:eventtype" value="paused" />
  <id key="opc:userid" value="1000000001" />
  <id key="opc:clientid" value="MILANKODOROOF2A" />
  <boolean key="opc:systemEvent" value="false" />
</event>
<event>
  <string key="opc:caseid" value="620c9847-a6a4-44f3-b335-91f8f513dd12" />
  <date key="time:timestamp" value="2015-10-21T18:42:23.03" />
  <id key="opc:eventid" value="c9589cea-33bb-421a-82ad-2ec277b4047f" />
  <string key="opc:sourcetype" value="case" />
  <string key="opc:sourceid" value="Reha Herr Schmidt" />
  <string key="opc:eventtype" value="closed" />
  <string key="opc:businessProcessType" value="Reha Management" />
  <id key="opc:userid" value="1000000001" />
  <id key="opc:clientid" value="MILANKODOROOF2A" />
  <boolean key="opc:systemEvent" value="false" />
</event>
</trace>
</log>

```

```
<log xes.version="1.4" xes.features="nested-attributes" openxes.version="1.8"
xmlns="http://www.xes-standard.org/">
  <extension name="Concept" prefix="concept" uri="http://www.xes-
standard.org/concept.xesext" />
  <extension name="Time" prefix="time" uri="http://www.xes-standard.org/time.xesext" />
  <extension name="ID" prefix="identity" uri="http://www.xes-standard.org/identity.xesext"
/>
  <extension name="OPC" prefix="opc" uri="http://www.paluno.de/uni-due/opc.xesext" />
  <global scope="trace">
    <string key="opc:caseid" value="string" />
  </global>
  <global scope="event">
    <string key="opc:eventid" value="string" />
    <string key="opc:sourcetype" value="string" />
    <string key="opc:sourceid" value="string" />
    <string key="opc:eventtype" value="string" />
    <string key="opc:userid" value="string" />
    <string key="opc:clientid" value="string" />
  </global>
  <trace>
    <string key="opc:caseid" value="620c9847-a6a4-44f3-b335-91f8f513dd12" />
    <event xmlns="">
      <string key="opc:caseid" value="620c9847-a6a4-44f3-b335-91f8f513dd12" />
      <date key="time:timestamp" value="2015-10-21T18:39:47.14" />
      <id key="opc:eventid" value="ce0f5912-5ac6-460b-924a-8ce703ae66b4" />
      <string key="opc:sourcetype" value="case" />
      <string key="opc:sourceid" value="Reha Herr Schmidt" />
      <string key="opc:eventtype" value="started" />
      <string key="opc:businessProcessType" value="Reha Management" />
      <id key="opc:userid" value="1000000001" />
      <id key="opc:clientid" value="MILANKODOR00F2A" />
      <boolean key="opc:systemEvent" value="false" />
    </event>
  </trace>
  <trace>
    <event xmlns="">
      <string key="opc:caseid" value="620c9847-a6a4-44f3-b335-91f8f513dd12" />
      <date key="time:timestamp" value="2015-10-21T18:39:47.14" />
      <id key="opc:eventid" value="6996368c-37c3-4150-b1c5-1cc21d6892db" />
      <string key="opc:sourcetype" value="agenda" />
      <string key="opc:sourceid" value="Reha Herr Schmidt" />
      <string key="opc:eventtype" value="started" />
      <id key="opc:userid" value="1000000001" />
      <id key="opc:clientid" value="MILANKODOR00F2A" />
      <boolean key="opc:systemEvent" value="false" />
    </event>
    <event>
      <string key="opc:caseid" value="620c9847-a6a4-44f3-b335-91f8f513dd12" />
      <date key="time:timestamp" value="2015-10-21T18:39:54.23" />
      <id key="opc:eventid" value="3a6bf41d-ce44-431a-88ec-6599f30832b5" />
      <string key="opc:sourcetype" value="item" />
      <string key="opc:sourceid" value="Heilverlaufsplan" />
      <string key="opc:eventtype" value="added" />
      <string key="opc:parent" value="Reha Herr Schmidt/" />
      <id key="opc:userid" value="1000000001" />
      <id key="opc:clientid" value="MILANKODOR00F2A" />
      <boolean key="opc:systemEvent" value="false" />
    </event>
    <event>
      <string key="opc:caseid" value="620c9847-a6a4-44f3-b335-91f8f513dd12" />
      <date key="time:timestamp" value="2015-10-21T18:40:00.66" />
      <id key="opc:eventid" value="85be2c8e-d460-496d-a61a-bf3787df1853" />
      <string key="opc:sourcetype" value="item" />
      <string key="opc:sourceid" value="Roentgenaufnahmen" />
      <string key="opc:eventtype" value="added" />
      <string key="opc:parent" value="Reha Herr Schmidt/" />
      <id key="opc:userid" value="1000000001" />
      <id key="opc:clientid" value="MILANKODOR00F2A" />
      <boolean key="opc:systemEvent" value="false" />
    </event>
    <event>
      <string key="opc:caseid" value="620c9847-a6a4-44f3-b335-91f8f513dd12" />
      <date key="time:timestamp" value="2015-10-21T18:40:09.86" />
      <id key="opc:eventid" value="b44947c1-54be-4105-b01b-2335aa92a7f0" />
      <string key="opc:sourcetype" value="artifact" />
      <string key="opc:sourceid" value="Diagnose.txt" />
      <string key="opc:eventtype" value="added" />
      <string key="opc:workspace" value="Reha Herr Schmidt/Heilverlaufsplan/" />
    </event>
  </trace>
</log>
```

10.3 SEMANTISCH VERKNÜPFTES EREIGNISPROTOKOLL (AUSZUG)

```

@prefix adcm: <http://paluno.uni-due.de/adcm#> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix owl: <http://www.w3.org/2002/07/owl#> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix event: <http://paluno.uni-due.de/opc/example/Event/> .
@prefix j.0: <http://paluno.uni-due.de/adcm#> .
@prefix case: <http://paluno.uni-due.de/opc/example/> .

event:b1734ecc-fee6-49aa-9c3d-8c077835b186
  a j.0:AgendaItemSelected ;
  j.0:eventOnAgendaItem <http://paluno.uni-due.de/opc/example/52f9fa2e-28be-4114-90ed-
ac650721897c/00001121/General_Lab_Clinical_Chemistry/aannname_laboratoriumonderzoek/> ;
  j.0:nextEvent event:fd3e6f4a-e092-4dbd-b1dc-bed6f93ad555 ;
  j.0:timestamp "2008-02-13T23:00:00Z"^^xsd:dateTime .

event:262ad237-ab30-4840-a17a-7a5530ec66c2
  a j.0:AgendaStarted ;
  j.0:eventOnAgendaItem <http://paluno.uni-due.de/opc/example/52f9fa2e-28be-4114-90ed-
ac650721897c/00001121/> ;
  j.0:nextEvent event:4c4c8f24-8ab5-4913-b25d-2a5e36737fda ;
  j.0:timestamp "2008-01-13T23:14:24Z"^^xsd:dateTime .

event:8da2f6bb-bcaf-4815-b088-785da151b81b
  a j.0:AgendaItemAdded ;
  j.0:eventOnAgendaItem <http://paluno.uni-due.de/opc/example/52f9fa2e-28be-4114-90ed-
ac650721897c/00001121/General_Lab_Clinical_Chemistry/aannname_laboratoriumonderzoek/> ;
  j.0:nextEvent event:b1734ecc-fee6-49aa-9c3d-8c077835b186 ;
  j.0:timestamp "2008-02-13T23:00:00Z"^^xsd:dateTime .

event:1edc8f9b-add1-47cc-a852-7c15eaec7275
  a j.0:AgendaItemAdded ;
  j.0:eventOnAgendaItem <http://paluno.uni-due.de/opc/example/52f9fa2e-28be-4114-90ed-
ac650721897c/00001121/General_Lab_Clinical_Chemistry/bilirubine_-_totaal/> ;
  j.0:nextEvent event:048765df-5761-484c-8c26-4845f1b6985f ;
  j.0:timestamp "2008-02-13T23:00:00Z"^^xsd:dateTime .

event:21175a66-f375-4a31-a7ea-9f727a3c183d
  a j.0:CaseStarted ;
  j.0:eventOnCase <http://paluno.uni-due.de/opc/example/52f9fa2e-28be-4114-90ed-ac650721897c/> ;
  j.0:nextEvent event:262ad237-ab30-4840-a17a-7a5530ec66c2 ;
  j.0:timestamp "2008-01-13T23:14:24Z"^^xsd:dateTime .

event:5f89aed5-3ccf-4432-ae18-e38f231d4756
  a j.0:AgendaItemSelected ;
  j.0:eventOnAgendaItem <http://paluno.uni-due.de/opc/example/52f9fa2e-28be-4114-90ed-
ac650721897c/00001121/General_Lab_Clinical_Chemistry/bilirubine_-_geconjugueerd/> ;
  j.0:nextEvent event:1edc8f9b-add1-47cc-a852-7c15eaec7275 ;
  j.0:timestamp "2008-02-13T23:00:00Z"^^xsd:dateTime .

event:ad16a363-5a9e-46a0-82ac-f8f2bc6ace75
  a j.0:AgendaItemAdded ;
  j.0:eventOnAgendaItem <http://paluno.uni-due.de/opc/example/52f9fa2e-28be-4114-90ed-
ac650721897c/00001121/General_Lab_Clinical_Chemistry/bilirubine_-_geconjugueerd/> ;
  j.0:nextEvent event:5f89aed5-3ccf-4432-ae18-e38f231d4756 ;
  j.0:timestamp "2008-02-13T23:00:00Z"^^xsd:dateTime .

event:fd3e6f4a-e092-4dbd-b1dc-bed6f93ad555
  a j.0:AgendaItemSelected ;
  j.0:eventOnAgendaItem <http://paluno.uni-due.de/opc/example/52f9fa2e-28be-4114-90ed-
ac650721897c/00001121/General_Lab_Clinical_Chemistry/aannname_laboratoriumonderzoek/> ;
  j.0:nextEvent event:971d2f79-a20c-4c34-8ca5-a870dd8a765e ;
  j.0:timestamp "2008-02-13T23:00:00Z"^^xsd:dateTime .

event:4c4c8f24-8ab5-4913-b25d-2a5e36737fda
  a j.0:AgendaItemAdded ;
  j.0:eventOnAgendaItem <http://paluno.uni-due.de/opc/example/52f9fa2e-28be-4114-90ed-
ac650721897c/00001121/General_Lab_Clinical_Chemistry/> ;
  j.0:nextEvent event:8da2f6bb-bcaf-4815-b088-785da151b81b ;
  j.0:timestamp "2008-02-13T23:00:00Z"^^xsd:dateTime .

event:971d2f79-a20c-4c34-8ca5-a870dd8a765e
  a j.0:AgendaItemSelected ;
  j.0:eventOnAgendaItem <http://paluno.uni-due.de/opc/example/52f9fa2e-28be-4114-90ed-
ac650721897c/00001121/General_Lab_Clinical_Chemistry/aannname_laboratoriumonderzoek/> ;
  j.0:nextEvent event:ad16a363-5a9e-46a0-82ac-f8f2bc6ace75 ;
  j.0:timestamp "2008-02-13T23:00:00Z"^^xsd:dateTime .

```

10.4 KONTEXTTERM-VEKTOREN (AUSZUG)

10.4.1 SZENARIO 1

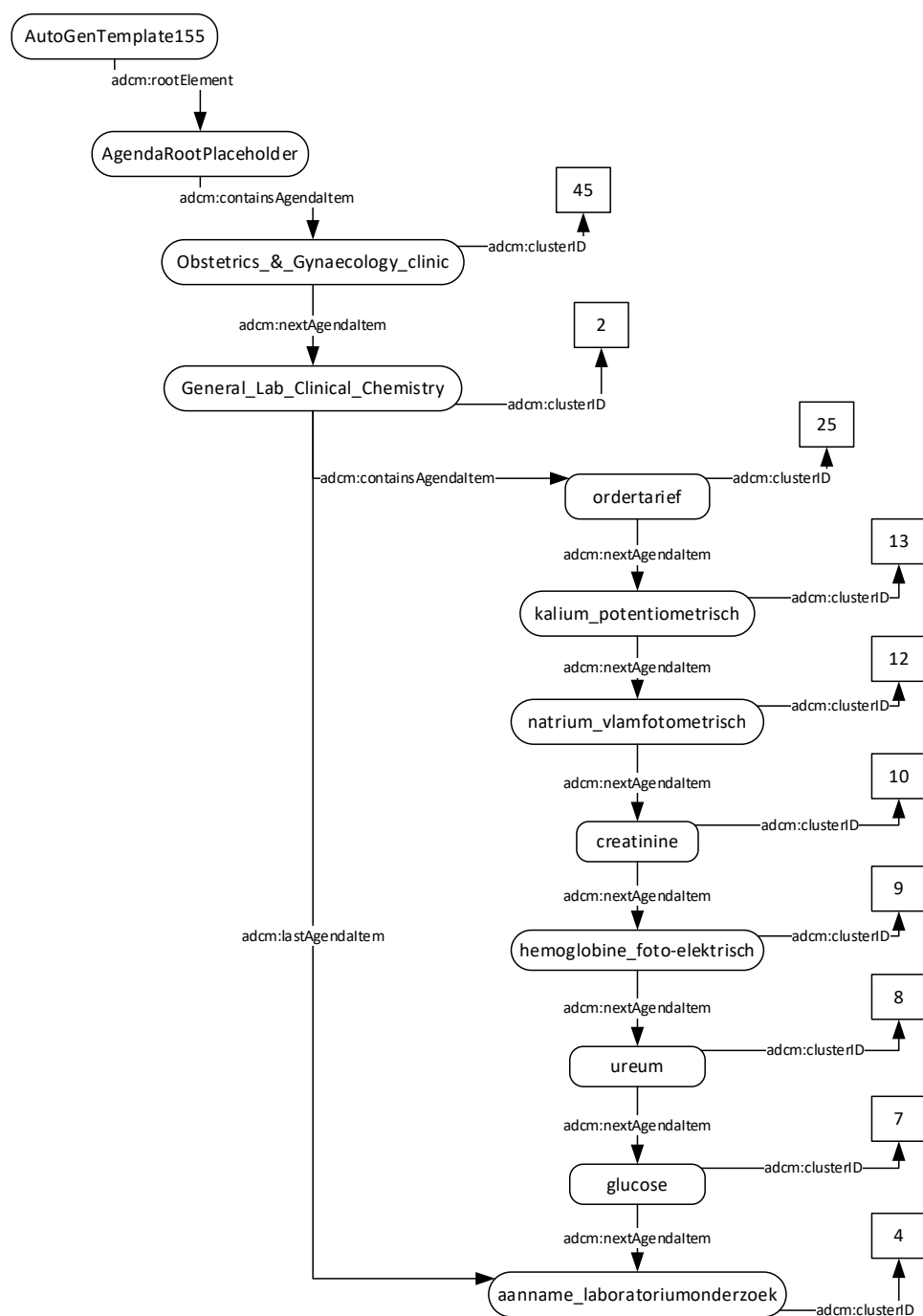
	12 von 51 Kontextbegriffen (mit Angabe der Document Frequency, DF)											
Name des Artefakts	fraktur (DF=2)	abheilung (DF=1)	operation (DF=1)	erlitt (DF=2)	sturz (DF=3)	fruhmobilisierung	morphin (DF=2)	ibuprofen (DF=1)	reha (DF=3)	fetthaltig (DF=1)	physiotherapi (DF=4)	radiuskopch (DF=1)
Fall 1A												
Diagnose	1	0	0	1	1	0	0	0	0	0	0	0
Medika- tion	0	0	0	0	0	1	1	0	0	0	0	0
Heilver- laufsplan	0	0	0	0	0	0	0	0	1	0	1	0
Fall 1B												
Diagnose	0	0	0	1	1	0	0	0	0	0	0	0
Medika- tion	0	0	0	0	0	1	1	0	0	0	0	0
Heilver- laufsplan	0	0	0	0	0	0	0	0	1	0	1	0
Fall 1C												
Diagnose	1	0	0	0	1	0	0	0	0	0	0	1
Medika- tion	0	0	0	0	0	0	0	1	0	0	0	0
Heilver- laufsplan	0	0	0	0	0	0	0	0	1	0	1	0
Fall 1D												
Diagnose	0	0	0	0	0	0	0	0	0	0	0	0

Medika- tion	0	1	0	0	0	0	0	0	0	1	0	0
Heilver- laufsplan	0	0	1	0	0	0	0	0	0	0	1	0

10.4.2 SZENARIO 2

	12 von 791 Kontextbegriffe (mit Angabe der Document Frequency, DF)											
Name des Artefakts	erstbehandelnd (DF=1)	insof (DF=1)	regelwidrig (DF=1)	roll (DF=1)	knoch (DF=3)	zeig (DF=1)	komplikation (DF=2)	gelockert (DF=1)	rotationsabweichung (DF=2)	verlief (DF=2)	unmittelbar (DF=1)	klinisch (DF=3)
Fall 1A												
Fallbesch- reibung	0	0	0	0	0	0	0	0	0	0	0	0
Gutachten	0	0	0	0	0	0	2	0	0	1	1	0
Fall 1B												
Fallbesch- reibung	0	0	0	0	1	0	0	0	1	0	0	1
Gutachten	1	0	0	1	3	1	0	0	4	0	0	1
Fall 1C												
Fallbesch- reibung	0	0	0	0	1	0	0	0	0	0	0	1
Gutachten	0	2	1	0	0	0	0	1	0	0	0	0
Fall 1D												
Fallbesch- reibung	0	0	0	0	0	0	1	0	0	1	0	0
Gutachten	0	0	0	0	0	0	0	0	0	0	0	0

10.5 BEISPIEL-TEMPLATE (IN GRAPHSTRUKTUR)



10.6 XES-ERWEITERUNG

```
<xesextension name="adcm" prefix="adcm" uri="http://.../adcm.xesext">
  <trace>
    <string key="caseid">      <!-- Identifies the case -->
    </string>
  </trace>
  <event>
    <string key="eventid">    <!-- Identifies the event -->
    </string>
    <string key="eventtype"> <!-- Type of the event according to the source,
                             e.g. started, paused, resumed, added, deleted -->
    </string>
    <string key="sourceid">   <!-- The name of the artifact, agenda item etc. -->
    </string>
    <string key="sourcetype"> <!-- One of {case, agenda, agendaitem, artifact} -->
    </string>
    <string key="userid">     <!-- Refers to the case manager -->
    </string>
  </event>
</xesextension>
```

11 LITERATURVERZEICHNIS

A. K. A. de Medeiros; B. F. van Dongen; W. M. P. van der Aalst; A. J. M. M. Weijters (2004): Process mining: Extending the α -algorithm to mine short loops. In: Eindhoven University of Technology (Hg.): BETA Working Paper Series. Eindhoven (WP 113).

Aalst, Wil M.P. van der; Weske, Mathias; Grünbauer, Dolf (2005): Case handling: a new paradigm for business process support. In: *DKE* 53 (2), S. 129–162. DOI: 10.1016/j.datak.2004.07.003.

Aamodt, Agnar; Plaza, Enric (1994): Case-based reasoning: foundational issues, methodological variations, and system approaches. In: *AI Commun* 7 (1), S. 39–59.

Akar, Ercan (2014): Process Mining für semi-strukturierte Geschäftsprozesse. Bachelorarbeit. Universität Duisburg-Essen, Essen. paluno.

Armour, Phillip G. (2000): The five orders of ignorance. In: *Commun. ACM* 43 (10), S. 17–20. DOI: 10.1145/352183.352194.

Artini, Michele; Atzori, Claudio; Bardi, Alessia; La Bruzzo, Sandro; Manghi, Paolo (2014): TagTick: A Tool for Annotation Tagging over Solr Indexes. In: Proceedings of the 14th ACM/IEEE-CS Joint Conference on Digital Libraries. Piscataway, NJ, USA: IEEE Press (JCDL '14), S. 407–408. Online verfügbar unter <http://dl.acm.org/citation.cfm?id=2740769.2740840>.

B.F. van Dongen (2015): BPI Challenge 2015 Municipality 2. Unter Mitarbeit von Eindhoven University of Technology.

Baeza-Yates, Ricardo (2015): SIGIR'15. Proceedings of the 38th ACM SIGIR Conference on Research and Development in Information Retrieval, August 9 - 13, 2015, Santiago, Chile; [including papers from the SIGIR Symposium on Information Retrieval in Practice (SIRIP), the SIGIR industry track]. 1 CD-ROM. New York, NY: ACM.

Bass, Len; Clements, Paul; Kazman, Rick (2013): Software architecture in practice. 3rd ed. Upper Saddle River, NJ: Addison-Wesley (SEI series in software engineering).

Bederson, Benjamin B.; Shneiderman, Ben; Wattenberg, Martin (2002): Ordered and quantum treemaps: Making effective use of 2D space to display hierarchies. In: *ACM Trans. Graph.* 21 (4), S. 833–854. DOI: 10.1145/571647.571649.

Beierle, Christoph; Kern-Isberner, Gabriele (2008): Methoden wissensbasierter Systeme. Grundlagen, Algorithmen, Anwendungen. 4., verb. Aufl. Braunschweig [u.a.]: Vieweg + Teubner (Studium).

Benner, Marian; Book, Matthias; Brückmann, Tobias; Gruhn, Volker; Richter, Thomas; Seyhan, Sema (2012): Managing and Tracing the Traversal of Process Clouds with Templates, Agendas and Artifacts. In: BPM 2012.

Benner-Wickner, Marian; Book, M.; Brückmann, Tobias; Gruhn, V. (2015a): Process Mining for Knowledge-intensive Business Processes. In: Proceedings of the 15th International Conference on Knowledge Technologies and Data-driven Business: ACM (i-KNOW '15).

Benner-Wickner, Marian; Book, Matthias; Bruckmann, Tobias; Gruhn, Volker (2014a): Examining Case Management Demand Using Event Log Complexity Metrics. In: 2014 IEEE 18th International Enterprise Distributed Object Computing Conference Workshops and Demonstrations (EDOCW). Ulm, Germany, S. 108–115.

Benner-Wickner, Marian; Book, Matthias; Brückmann, Tobias; Gruhn, Volker (2013): Cloud Storage of Artifact Annotations to Support Case Managers in Knowledge-Intensive Business Processes. In: WISE 2012, Bd. 7652. Berlin, Heidelberg: Springer (LNCS), S. 92–104.

Benner-Wickner, Marian; Book, Matthias; Brückmann, Tobias; Gruhn, Volker (2014b): Execution Support for Agenda-Driven Case Management. In: ACM SAC '14.

Benner-Wickner, Marian; Book, Matthias; Brückmann, Tobias; Gruhn, Volker (2016): Adapting Case Management Techniques to Achieve Software Process Flexibility. In: Marco Kuhrmann, Münch Münch, Ita Richardson, Andreas Rausch und He Zhang (Hg.): Managing software process evolution. 1st ed. 2016. [S.l.]: Springer.

Benner-Wickner, Marian; Koop, Wilhelm; Book, Matthias; Gruhn, Volker (2015b): Supporting Adaptive Case Management Through Semantic Web Technologies. In: 4th International Workshop on Adaptive Case Management and other Non-workflow Approaches to BPM (AdaptiveCM'15): Springer (Lecture Notes in Business Information Processing (LNBIP)).

Berners-Lee, T.; Fielding, R.; Masinter, L. (2005): Uniform Resource Identifier (URI). Generic Syntax. RFC Editor (3986). Online verfügbar unter <http://www.rfc-editor.org/rfc/rfc3986.txt>.

- Berners-Lee, Tim (2006): Linked Data. W3C ((<http://www.w3.org/DesignIssues/LinkedData.html>)). Online verfügbar unter <http://www.w3.org/DesignIssues/LinkedData.html>, zuletzt geprüft am 07.05.2015.
- Berry, Michael J. A.; Linoff, Gordon (2000): Mastering data mining. The art and science of customer relationship management. New York: Wiley.
- Bloebaum, Trude H.; Johnsen, Frank T. (2015): Evaluating publish/subscribe approaches for use in tactical broadband networks. In: MILCOM 2015 - 2015 IEEE Military Communications Conference. Tampa, FL, USA, S. 605–610.
- Buijs, Joos C. A. M.; Dongen, Boudewijn F.; Aalst, Wil M. P. van der (2012): On the Role of Fitness, Precision, Generalization and Simplicity in Process Discovery. In: OTM 2012, Bd. 7565. Berlin, Heidelberg: Springer (LNCS), S. 305–322.
- Burattin, Andrea; Sperduti, Alessandro (2011): PLG: A Framework for the Generation of Business Process Models and Their Execution Logs. In: Will van der Aalst, John Mylopoulos, Norman M. Sadeh, Michael J. Shaw, Clemens Szyperski, Michael Zur Muehlen und Jianwen Su (Hg.): Business Process Management Workshops, Bd. 66. Berlin, Heidelberg: Springer Berlin Heidelberg (Lecture Notes in Business Information Processing), S. 214–219.
- Buschmann, Frank (1996): Pattern-oriented software architecture. A system of patterns, volume 1. Chichester, New York: Wiley.
- Business Process Management Workshops (2013). Berlin, Heidelberg: Springer (LNBIP).
- Chehreghani, Mostafa Haghiri (2011): Efficiently Mining Unordered Trees. In: 2011 IEEE 11th International Conference on Data Mining (ICDM). Vancouver, BC, Canada, S. 111–120.
- Chen, Tao; Zhang, Yongjuan; Zhang, Shen; Chen, Chengcai; Chen, Heng (2012): Building semantic information search platform with extended Sesame framework. In: Valentina Presutti und H. Sofia Pinto (Hg.): the 8th International Conference on Semantic Systems. Graz, Austria, S. 193.
- Chi, Y.; Yang, Y.; Muntz, R. R. (2003): Indexing and mining free trees. In: Third IEEE International Conference on Data Mining. Melbourne, FL, USA, 19-22 Nov. 2003, S. 509–512.
- Chi, Yun; Yang, Yirong; Muntz, Richard R. (2005): Canonical forms for labelled trees and their applications in frequent subtree mining. In: *Knowl Inf Syst* 8 (2), S. 203–234. DOI: 10.1007/s10115-004-0180-7.
- Chi, Yun; Yang, Yirong; Xia, Yi; Muntz, Richard R. (2004): CMTreeMiner: Mining Both Closed and Maximal Frequent Subtrees. In: Advances in Knowledge Discovery and Data Mining, Bd. 3056. Berlin, Heidelberg: Springer (LNCS), S. 63–73.

Curino, Carlo; Moon, Hyun Jin; Tanca, Letizia; Zaniolo, Carlo: Schema Evolution in Wikipedia - Toward a Web Information System Benchmark Enterprise Information Systems, Volume DISI, Barcelona, Spain, June 12-16, 2008, S. 323–332.

de Medeiros, A. K. A.; Weijters, A. J. M. M.; van der Aalst, W. M. P. (2007): Genetic process mining: an experimental evaluation. In: *Data Min Knowl Disc* 14 (2), S. 245–304. DOI: 10.1007/s10618-006-0061-7.

de Medeiros, Ana Karla Alves; Guzzo, Antonella; Greco, Gianluigi; van der Aalst, Wil M. P.; Weijters, A. J. M. M.; van Dongen, Boudewijn F.; Saccà, Domenico (2008): Process Mining Based on Clustering: A Quest for Precision. In: David Hutchison, Takeo Kanade, Josef Kittler, Jon M. Kleinberg, Friedemann Mattern, John C. Mitchell et al. (Hg.): Business Process Management Workshops, Bd. 4928. Berlin, Heidelberg: Springer Berlin Heidelberg (Lecture Notes in Computer Science), S. 17–29.

Demey, Yan Tang; Panetto, Herve (2013): On the move to meaningful internet systems: OTM 2013 Workshops. Confederated International Workshops: OTM Academy, OTM Industry Case Studies Program, ACM, EI2N, ISDE, META4eS, ORM, SeDeS, SINCOM, SMS, and SOMOCO 2013, Graz, Austria, September 9-13, 2013, Proceedings (LNCS sublibrary. SL 3, Information systems and application, incl. Internet/Web and HCI, 8186).

DGUV (2014): Das Reha-Management der Deutschen Gesetzlichen Unfallversicherung. Handlungsleitfaden. Hg. v. Deutsche Gesetzliche Unfallversicherung e.V. Online verfügbar unter <http://www.dguv.de/medien/inhalt/rehabilitation/documents/handlungsleitfaden.pdf>, zuletzt geprüft am 13.07.2016.

Drago, Idilio; Mellia, Marco; M. Munafo, Maurizio; Sperotto, Anna; Sadre, Ramin; Pras, Aiko (2012): Inside dropbox. In: John Byers, Jim Kurose, Ratul Mahajan und Alex C. Snoeren (Hg.): Proceedings of the 2012 ACM conference on Internet measurement. Boston, Massachusetts, USA, S. 481.

Drucker, Peter F. (1992): The age of discontinuity. Guidelines to our changing society. New Brunswick (U.S.A.): Transaction Pubs.

Dumas, Marlon (2013): Fundamentals of business process management. Berlin, New York: Springer.

Dunkel, Jürgen; Eberhart, Andreas; Fischer, Stefan; Kleiner, Carsten; Koschel, Arne (2008): Systemarchitekturen für Verteilte Anwendungen. Client-Server, Multi-Tier, SOA, Event Driven Architectures, P2P, Grid, Web 2.0. 1. Aufl. München: Hanser, Carl.

- Ermolayev, Vadim; Jentzsch, Eyck; Sohnius, Richard; Warren, Paul W. (2008): ACTIVE Technologies in IC Design: Acquiring, Articulating, and Sharing Design Process Knowledge. A White Paper. Hg. v. ACTIVE CONSORTIUM, zuletzt geprüft am 02.10.2015.
- Eugster, Patrick Th.; Felber, Pascal A.; Guerraoui, Rachid; Kermarrec, Anne-Marie (2003): The many faces of publish/subscribe. In: *ACM Comput. Surv.* 35 (2), S. 114–131. DOI: 10.1145/857076.857078.
- Fellbaum, Christiane (2012): WordNet. In: Carol A. Chapelle (Hg.): *The Encyclopedia of Applied Linguistics*. Oxford, UK: Blackwell Publishing Ltd.
- Fernandes, Joel L.; Lopes, Ivo C.; Rodrigues, Joel J. P. C.; Ullah, Sana (2013): Performance evaluation of RESTful web services and AMQP protocol. In: 2013 Fifth International Conference on Ubiquitous and Future Networks (ICUFN). DA NANG, Vietnam, S. 810–815.
- Fischer, Layna (Hg.) (2015): *Thriving on adaptability. Best practices for knowledge workers*. Lighthouse Point, FL: Future Strategies Inc.
- Fischer, Layna; Miers, Derek (2013): *Empowering knowledge workers. New ways to leverage case management (BPM and workflow handbook series)*.
- Flock, Marvin (2014): *Konzeption und Entwicklung einer Treemap-Benutzerschnittstelle für Case-Management-Tools*. Bachelorarbeit. Universität Duisburg-Essen, Essen. paluno.
- Fraktionen SPD und BÜNDNIS 90/DIE GRÜNEN (2003): Entwurf eines Vierten Gesetzes für moderne Dienstleistungen am Arbeitsmarkt. BT-Drs 16/1516.
- Gadatsch, Andreas (2010): *Grundkurs Geschäftsprozess-Management. Methoden und Werkzeuge für die IT-Praxis; eine Einführung für Studenten und Praktiker*. 6., aktualisierte Aufl. Wiesbaden: Vieweg + Teubner (Studium).
- Gamma, Erich (1995): *Design patterns. Elements of reusable object-oriented software*. Reading, Mass.: Addison-Wesley (Addison-Wesley professional computing series).
- Gini, Corrado (1912): *Variabilità e Mutuabilità. Contributo allo Studio delle Distribuzioni e delle Relazioni Statistiche*. Bologna.
- Greco, G.; Guzzo, A.; Pontieri, L.; Sacca, D. (2006): Discovering expressive process models by clustering log traces. In: *IEEE Trans. Knowl. Data Eng.* 18 (8), S. 1010–1027. DOI: 10.1109/TKDE.2006.123.
- Gruhn, Volker; Laue, Ralf (2006): Complexity metrics for business process models. In: 9th international conference on business information systems (BIS 2006), Bd. 85. Citeseer, S. 1–12.

- Günther, Christian W.; van der Aalst, Wil M. P. (2007): Fuzzy Mining – Adaptive Process Simplification Based on Multi-perspective Metrics. In: BPM 2007, Bd. 4714. Berlin, Heidelberg: Springer (LNCS), S. 328–343.
- Gupta, Shaifali (2007): Workflow and Process Mining in Healthcare. Master Thesis. Eindhoven University of Technology, Eindhoven. Department of Mathematics and Computer Science.
- Hadzic, Fedja; Tan, Henry; Dillon, Tharam S. (2010): Mining of Data with Complex Structures. Berlin, Heidelberg: Springer (Studies in Computational Intelligence, 333).
- Hamming, Richard W. (1950): Error detecting and error correcting codes. In: *Bell System technical journal* 29 (2), S. 147–160.
- Hardesty, Juliet L.; Greene, Courtney (2013): MOOD-lighting. In: J. Stephen Downie, Robert H. McDonald, Timothy W. Cole, Robert Sanderson und Frank Shipman (Hg.): the 13th ACM/IEEE-CS joint conference on Digital libraries. Indianapolis, Indiana, USA, S. 409.
- Hartigan, John A. (1975): Clustering algorithms. New York: Wiley (A Wiley publication in applied statistics).
- Heisig, Peter (1999): Geschäftsprozeßorientiertes Wissensmanagement. In: *Total Business Knowledge Management - Wissensmanagement verändert Unternehmen*, S. 27–42.
- Henry, S.; Kafura, D. (1981): Software Structure Metrics Based on Information Flow. In: *IEEE Trans. Software Eng. (IEEE Transactions on Software Engineering)* SE-7 (5), S. 510–518. DOI: 10.1109/TSE.1981.231113.
- Hido, S.; Kawano, H. (2005): AMIOT: Induced Ordered Tree Mining in Tree-Structured Databases. In: Fifth IEEE International Conference on Data Mining (ICDM'05). Houston, TX, USA, 27-30 Nov. 2005, S. 170–177.
- Hohpe, Gregor; Woolf, Bobby (2004): Enterprise integration patterns. Designing, building, and deploying messaging solutions. Boston: Addison-Wesley (The Addison-Wesley signature series).
- IEEE (2014): 2014 EDOC Workshops.
- Jiang, Chuntao; Coenen, Frans; Zito, Michele (2013): A survey of frequent subgraph mining algorithms. In: *The Knowledge Engineering Review* 28 (01), S. 75–105. DOI: 10.1017/S0269888912000331.
- Jiménez, Aída; Berzal, Fernando; Cubero, Juan-Carlos (2008): Mining Induced and Embedded Subtrees in Ordered, Unordered, and Partially-Ordered Trees. In: Aijun An, Stan Matwin, Zbigniew W. Raś und Dominik Ślęzak (Hg.): Foundations of Intelligent Systems, Bd. 4994. Berlin, Heidelberg: Springer Berlin Heidelberg (Lecture Notes in Computer Science), S. 111–120.

- Jones, M.; Bradley, J.; Sakimura, N. (2015): JSON Web Token (JWT): RFC Editor.
- Kalinowski, Alexander (2013): Agenda mining in the context of agenda-driven Case Management. Seminararbeit. Universität Duisburg-Essen, Essen. paluno.
- Kelley, Robert Earl (1985): The gold-collar worker. Harnessing the brainpower of the new workforce. Reading, Mass.: Addison-Wesley.
- Köhncke, Benjamin; Siehndel, Patrick; Balke, Wolf-Tilo (2013): Bridging the Gap – Using External Knowledge Bases for Context-Aware Document Retrieval. In: David Hutchison, Takeo Kanade, Josef Kittler, Jon M. Kleinberg, Friedemann Mattern, John C. Mitchell et al. (Hg.): Digital Libraries: Social Media and Community Networks, Bd. 8279. Cham: Springer International Publishing (Lecture Notes in Computer Science), S. 11–20.
- Koop, Wilhelm (2014): Entwicklung einer Ontologie zur Gewinnung von Prozesswissen im Agenda-Driven Case Management. Masterarbeit. Universität Duisburg-Essen, Essen. paluno.
- Kraft, Reiner; Chang, Chi Chao; Maghoul, Farzin; Kumar, Ravi (2006): Searching with context. In: Leslie Carr, David de Roure, Arun Iyengar, Carole Goble und Mike Dahlin (Hg.): the 15th international conference on World Wide Web. Edinburgh, Scotland, S. 477.
- Kraft, Reiner; Maghoul, Farzin; Chang, Chi Chao (2005): Y!Q. In: Otthein Herzog, Hans-Jörg Schek, Norbert Fuhr, Abdur Chowdhury und Wilfried Teiken (Hg.): Proceedings of the 14th ACM international conference on Information and knowledge management (CIKM). Bremen, Germany, S. 816.
- Krötzsch, Markus; Vrandečić, Denny; Völkel, Max; Haller, Heiko; Studer, Rudi (2007): Semantic Wikipedia. In: *Journal of Web Semantics* 5, S. 251–261.
- Krumke, Sven Oliver; Noltemeier, Hartmut (2012): Graphentheoretische Konzepte und Algorithmen. 3. Aufl. 2012. Wiesbaden: Vieweg+Teubner Verlag (Leitfäden der Informatik).
- Kunze, Claudia (2005): Semantische Relationstypen in GermaNet. In: Stefan Langer und Daniel Schnorbusch (Hg.): Semantik im Lexikon. Tübingen: G. Narr (Tübinger Beiträge zur Linguistik, 479), S. 161–178.
- Kurz, Matthias; Schmidt, Werner; Fleischmann, Albert; Lederer, Matthias (2015): Leveraging CMMN for ACM. In: Jens Ehlers und Bernhard Thalheim (Hg.): the 7th International Conference on Subject-Oriented Business Process Management. Kiel, Germany, S. 1–9.
- Lacic, Emanuel; Kowald, Dominik; Parra, Denis; Kahr, Martin; Trattner, Christoph (2014): Towards a Scalable Social Recommender Engine for Online Marketplaces: The Case of Apache Solr. In: Proceedings of the 23rd International Conference on World Wide Web. Republic and

Canton of Geneva, Switzerland: International World Wide Web Conferences Steering Committee (WWW '14 Companion), S. 817–822. Online verfügbar unter <http://dx.doi.org/10.1145/2567948.2579245>.

Leemans, Sander J. J.; Fahland, Dirk; van der Aalst, Wil M. P. (2013): Discovering Block-Structured Process Models from Event Logs - A Constructive Approach. In: PETRI NETS 2013, Bd. 7927. Berlin, Heidelberg: Springer (LNCS), S. 311–329.

Levenshtein, Vladimir I. (1966): Binary codes capable of correcting deletions, insertions, and reversals. In: Soviet physics doklady, Bd. 10, S. 707–710.

Li, Jiafei; Liu, Dayou; Yang, Bo (2007): Process Mining: Extending α -Algorithm to Mine Duplicate Tasks in Process Logs. In: Kevin Chen-Chuan Chang, Wei Wang, Lei Chen, Clarence A. Ellis, Ching-Hsien Hsu, Ah Chung Tsoi und Haixun Wang (Hg.): Advances in Web and Network Technologies, and Information Management, Bd. 4537. Berlin, Heidelberg: Springer Berlin Heidelberg (Lecture Notes in Computer Science), S. 396–407.

Lichtenegger, Wolfgang (2012): Methoden zur teilautomatischen Konstruktion von Ist-Prozessmodellen mittels Process Mining sowie zur Integration manuell konstruierter und automatisch generierter Ist-Prozessmodelle. Berlin: Logos.

Löcherbach, Peter (2009): Der Handlungsansatz Case Management. - oder: Warum Soziale Arbeit auch managen kann. In: Albert Mühlum (Hg.): Soziale Arbeit in Wissenschaft und Praxis. Festschrift für Wolf Rainer Wendt. Lage: Jacobs, S. 265–279.

Loeffeler, Thorsten; Striemer, Ruediger; Deiters, Wolfgang (1998): A framework for identification, classification and IT support of semi-structured business processes. In: *Knowledge and Process Management* 5 (1), S. 51–57. DOI: 10.1002/(SICI)1099-1441(199803)5:1<51::AID-KPM9>3.0.CO;2-L.

Luczak, Holger (1998): Konzepte und Methoden der Arbeitsanalyse. In: Holger Luczak (Hg.): Arbeitswissenschaft. Berlin, Heidelberg: Springer Berlin Heidelberg, S. 25–51.

Luzuriaga, Jorge E.; Perez, Miguel; Boronat, Pablo; Cano, Juan Carlos; Calafate, Carlos; Manzoni, Pietro (2015): Impact of mobility on Message Oriented Middleware (MOM) protocols for collaboration in transportation. In: 2015 IEEE 19th International Conference on Computer Supported Cooperative Work in Design (CSCWD). Calabria, Italy, S. 115–120.

Maercker, Andreas (Hg.) (2013): Posttraumatische Belastungsstörungen. Mit 40 Tabellen. 4., vollst. überarb. und aktualisierte Aufl. Berlin: Springer. Online verfügbar unter <http://dx.doi.org/10.1007/978-3-642-35068-9>.

- Makolm, Josef; Weiss, Silke; Ipsmiller, Doris (2008): DYONIPOS: Proactive support of knowledge workers. In: 2008 Third International Conference on Digital Information Management (ICDIM). London, United Kingdom, S. 286–291.
- Manning, Christopher D.; Raghavan, Prabhakar; Schütze, Hinrich (2008): Introduction to information retrieval. New York: Cambridge University Press.
- Mans, R. S.; Schonenberg, M. H.; Song, M.; van der Aalst, W. M. P.; Bakker, P. J. M. (2009): Application of Process Mining in Healthcare – A Case Study in a Dutch Hospital. In: BIOSTEC 2009, Bd. 25. Berlin, Heidelberg: Springer, S. 425–438.
- Mark, William; Simoudis, Evangelos; Hinkle, David (1996): Case-based reasoning: Expectations and results. In: David B. Leake (Hg.): Case-based reasoning. Experiences, lessons & future directions. Menlo Park, Calif., Cambridge, Mass.: AAAI Press; MIT Press, S. 269–294.
- McCauley, Dermot (2010): Achieving Agility. In: Keith D. Swenson (Hg.): Mastering the unpredictable: Meghan-Kiffer Press, S. 257–275.
- Motahari-Nezhad, Hamid Reza; Recker, Jan; Weidlich, Matthias (Hg.) (2015): Business Process Management. Cham: Springer International Publishing (Lecture Notes in Computer Science).
- Mühlum, Albert; Gödecker-Geenen, Norbert (2003): Soziale Arbeit in der Rehabilitation. München, Basel: Reinhardt (Soziale Arbeit im Gesundheitswesen, Bd. 1).
- Naisbitt, John (1984): Megatrends. Ten new directions transforming our lives. 1st [updated] paperback ed. New York: Warner Books.
- Navigli, Roberto; Velardi, Paola (2004): Learning Domain Ontologies from Document Warehouses and Dedicated Web Sites. In: *Computational Linguistics* 30 (2), S. 151–179. DOI: 10.1162/089120104323093276.
- Nijssen, Siegfried; Kok, Joost N. (2003): Efficient Discovery of Frequent Unordered Trees. In: Proceedings of the 1st International Workshop on Mining Graphs, Trees and Sequences, S. 55–64.
- Nisch, Stefan (2016): Evaluation eines ProcessMining-Algorithmus für Case Management Systeme. Bachelorarbeit. Universität Duisburg-Essen, Essen.
- Nonaka, I.; Takeuchi, H. (1997): Die Organisation des Wissens: wie japanische Unternehmen eine brachliegende Ressource nutzbar machen: Campus-Verl. Online verfügbar unter <http://books.google.com/books?id=IPZ0PQAACAAJ>.
- North, Klaus; Guldenberg, Stefan (2008): Produktive Wissensarbeit(er). Wiesbaden: Gabler.

Object Management Group (2013): Business Process Model and Notation (BPMN). 2.0.2. Aufl. (formal/2013-12-09). Online verfügbar unter <http://www.omg.org/spec/BPMN>.

Object Management Group (2014): Case Management Model and Notation (CMMN). 1.0. Aufl. (formal/2014-05-05). Online verfügbar unter <http://www.omg.org/spec/CMMN/1.0/>.

Paik, Jiaul H. (2013): A novel TF-IDF weighting scheme for effective ranking. In: Gareth J.F Jones, Páraic Sheridan, Diane Kelly, Maarten de Rijke und Tetsuya Sakai (Hg.): the 36th international ACM SIGIR conference. Dublin, Ireland, S. 343.

Peinl (2011): Unified Information Access. In: *Informatik Spektrum* 34/6, S. 594–597.

Pucher, Max J. (2010): The Elements Of Adaptive Case Management. In: Keith D. Swenson (Hg.): Mastering the unpredictable: Meghan-Kiffer Press, S. 89–134.

Rath, Andreas S. (2007): A Low-Level Based Task and Process Support Approach For Knowledge-Intensive Business Environments. In: DECIS '07: Doctoral Consortium held at ICEIS'07: INSTICC Press, S. 35–42.

Rath, Andreas S. (2010): User Interaction Context - Studying and Enhancing Automatic User Task Detection on the Computer Desktop via an Ontology-based User Interaction Context Model. Graz University of Technology.

Ratzer, Anne Vinter; Wells, Lisa; Lassen, Henry Michael; Laursen, Mads; Qvortrup, Jacob Frank; Stissing, Martin Stig et al. (2003): CPN Tools for Editing, Simulating, and Analysing Coloured Petri Nets. In: Gerhard Goos, Juris Hartmanis, Jan van Leeuwen, van der Aalst, Wil M. P und Eike Best (Hg.): Applications and Theory of Petri Nets 2003, Bd. 2679. Berlin, Heidelberg: Springer Berlin Heidelberg (Lecture Notes in Computer Science), S. 450–462.

W3C Recommendation, 25.02.2014: RDF 1.1 Concepts and Abstract Syntax. Online verfügbar unter <http://www.w3.org/TR/rdf11-concepts/>, zuletzt geprüft am 07.05.2015.

Reichert, Manfred; Dadam, Peter (2000): Geschäftsprozessmodellierung und Workflow-Management - Konzepte, Systeme und deren Anwendung. In: *Industrie Management* 16 (3), S. 23–27. Online verfügbar unter <http://dbis.eprints.uni-ulm.de/239/>.

Reichert, Manfred; Weber, Barbara (2012): Enabling flexibility in process-aware information systems. Challenges, methods, technologies. Heidelberg, New York: Springer.

Reijers, Hajo A.; Slaats, Tijs; Stahl, Christian (2013): Declarative Modeling–An Academic Dream or the Future for BPM? In: Florian Daniel, Jianmin Wang und Barbara Weber (Hg.): Business Process Management, Bd. 8094. Berlin, Heidelberg: Springer Berlin Heidelberg (Lecture Notes in Computer Science), S. 307–322.

Reiter, Raymond (1978): On Closed World Data Bases. In: Hervé Gallaire und Jack Minker (Hg.): Logic and Data Bases. Boston, MA: Springer US, S. 55–76.

Robertson, Stephen (2004): Understanding Inverse Document Frequency: On theoretical arguments for IDF. In: *Journal of Documentation* 60, S. 503–520. Online verfügbar unter <http://research.microsoft.com/apps/pubs/default.aspx?id=67744>.

Röder, Daniel (2010): JPA mit Hibernate. Java Persistence API in der Praxis. vollständig überarbeitete. Frankfurt am Main: entwickler.press.

Rosa, Marcello; Soffer, Pnina (2013): Business process management workshops. BPM 2012 International Workshops, Tallinn, Estonia, September 3, 2012. Revised papers. Berlin, New York: Springer (Lecture Notes in Business Information Processing, 132).

Rosing, Mark Von; Scheer, August-Wilhelm; Scheel, Henrik Von (Hg.) (2014): The complete business process handbook. Body of knowledge from process modeling to bpm.

Rozanski, Nick; Woods, Eoin (2012): Software systems architecture. Working with stakeholders using viewpoints and perspectives. 2nd ed. Upper Saddle River, NJ: Addison-Wesley.

Rozinat, A.; van der Aalst, W.M.P. (2008): Conformance checking of processes based on monitoring real behavior. In: *Information Systems* 33 (1), S. 64–95. DOI: 10.1016/j.is.2007.07.001.

Rozinat, Anne; de Medeiros, Ana Karla Alves; Günther, Christian W.; Weijters, A. J. M. M.; van der Aalst, Wil M. P. (2008): The Need for a Process Mining Evaluation Framework in Research and Practice. In: David Hutchison, Takeo Kanade, Josef Kittler, Jon M. Kleinberg, Friedemann Mattern, John C. Mitchell et al. (Hg.): Business Process Management Workshops, Bd. 4928. Berlin, Heidelberg: Springer Berlin Heidelberg (Lecture Notes in Computer Science), S. 84–89.

Rubin, Vladimir; Günther, Christian W.; van der Aalst, Wil M. P.; Kindler, Ekkart; van Dongen, Boudewijn F.; Schäfer, Wilhelm (2007): Process Mining Framework for Software Processes. In: David Hutchison, Takeo Kanade, Josef Kittler, Jon M. Kleinberg, Friedemann Mattern, John C. Mitchell et al. (Hg.): Software Process Dynamics and Agility, Bd. 4470. Berlin, Heidelberg: Springer Berlin Heidelberg (Lecture Notes in Computer Science), S. 169–181.

Rückert, Ulrich; Kramer, Stefan (2004): Frequent free tree discovery in graph data. In: Hisham M. Haddad, Andrea Omicini, Roger L. Wainwright und Lorie M. Liebrock (Hg.): the 2004 ACM symposium. Nicosia, Cyprus, S. 564.

Rychkova, Irina (2013): Towards Automated Support for Case Management Processes with Declarative Configurable Specifications. In: Business Process Management Workshops, Bd. 132. Berlin, Heidelberg: Springer (LNBIP), S. 65–76.

- Salton, G.; Wong, A.; Yang, C. S. (1975): A vector space model for automatic indexing. In: *Commun. ACM* 18 (11), S. 613–620. DOI: 10.1145/361219.361220.
- Schatten, Alexander; Inselkammer, Franz; Tjoa, A. Min (2003): System Integration and Unified Information Access using Question based Knowledge Management Strategies. In: Gabriele Kotsis, Stéphane Bressan und Ismail Khalil Ibrahim (Hg.): *iiWAS'2003 - The Fifth International Conference on Information Integration and Web-based Applications Services*, 15-17 September 2003, Jakarta, Indonesia: Austrian Computer Society (books@ocg.at, 170).
- Scheer, A. W.; Jost, W. (1996): Geschäftsprozessmodellierung innerhalb einer Unternehmensarchitektur. In: Gottfried Vossen (Hg.): *Geschäftsprozessmodellierung und Workflow-Management. Modelle, Methoden, Werkzeuge*. 1. Aufl. Bonn, Albany ([u.a.]: Internat. Thomson Publ. (Informatik Lehrbuch-Reihe), S. 29–46.
- Schermer, Franz J. (2014): *Lernen und Gedächtnis*. 5., überarb. und erw. Aufl. Stuttgart: Kohlhammer (Grundriß der Psychologie, 10).
- Schwarz, Sven; Abecker, Andreas; Maus, Heiko; Sintek, Michael (2001): Anforderungen an die Workflow-Unterstützung für wissensintensive Geschäftsprozesse. In: Workshop Geschäftsprozessorientiertes Wissensmanagement 2001, 01-02. Online verfügbar unter <http://www.dfki.uni-kl.de/schwarz/docs/SchwarzAbeckerMausSintek2001.pdf>.
- Shannon, C. E. (2001): A mathematical theory of communication. In: *SIGMOBILE Mob. Comput. Commun. Rev.* 5 (1), S. 3. DOI: 10.1145/584091.584093.
- Shasha, D.; Wang, J.T.L.; Sen Zhang (2004): Unordered tree mining with applications to phylogeny. In: *Proceedings. 20th International Conference on Data Engineering*. Boston, MA, USA, 30 March-2 April 2004, S. 708–719.
- Shneiderman, B. (1992): Tree visualization with Tree-maps: A 2-d space-filling approach. In: *ACM Transactions on Graphics (TOG)* (11), S. 92–99.
- Simperl, Elena; Thurlow, Ian; Warren, Paul; Dengler, Frank; Davies, John; Grobelnik, Marko et al. (2010): Overcoming Information Overload in the Enterprise: The Active Approach. In: *IEEE Internet Comput.* 14 (6), S. 39–46. DOI: 10.1109/MIC.2010.146.
- W3C Recommendation REC-soap12-part1-20070427, 27.04.2007: SOAP Version 1.2: Part 1: Messaging Framework (Second Edition). Online verfügbar unter <https://www.w3.org/TR/soap12/>, zuletzt geprüft am 24.06.2016.
- solidIT (2015): DB-Engines Ranking of Search Engines. solidIT consulting & software development gmbh. Online verfügbar unter <http://db-engines.com/en/ranking/search+engine>, zuletzt aktualisiert am 10.2015, zuletzt geprüft am 09.10.2015.

- Song, Minseok; van der Aalst, Wil M.P. (2008): Towards comprehensive support for organizational mining. In: *Decision Support Systems* 46 (1), S. 300–317. DOI: 10.1016/j.dss.2008.07.002.
- Spalzzi, Luca (2001): A Survey on Case-Based Planning. In: *Artificial Intelligence Review* 16 (1), S. 3–36. DOI: 10.1023/A:1011081305027.
- Starke, Gernot (2015): Effektive Softwarearchitekturen. Ein praktischer Leitfaden. 7., überarb. Aufl. München: Hanser.
- Swenson, Keith (2011): Taming the unpredictable Real World Adaptive Case Management. London: Future Strategies (Excellence in practice series).
- Swenson, Keith; von Rosing, Mark (2014): What is Business Process Management. In: Mark Von Rosing, August-Wilhelm Scheer und Henrik Von Scheel (Hg.): The complete business process handbook. Body of knowledge from process modeling to bpm, S. 79–88.
- Swenson, Keith D. (Hg.) (2010a): Mastering the unpredictable: Meghan-Kiffer Press.
- Swenson, Keith D. (2010b): The Nature of Knowledge Work. In: Keith D. Swenson (Hg.): Mastering the unpredictable: Meghan-Kiffer Press, S. 5–27.
- Swenson, Keith D. (2012): How knowledge workers get things done. London: Future Strategies (Excellence in practice series).
- Tan, Henry; Dillon, Tharam; Hadzic, Fedja; Chang, Elizabeth (2006a): Razor: mining distance-constrained embedded subtrees. In: Sixth IEEE International Conference on Data Mining - Workshops (ICDMW'06). Hong Kong, China, S. 8–13.
- Tan, Henry; Dillon, Tharam S.; Hadzic, Fedja; Chang, Elizabeth; Feng, Ling (2006b): IMB3-Miner: Mining Induced/Embedded Subtrees by Constraining the Level of Embedding. In: David Hutchison, Takeo Kanade, Josef Kittler, Jon M. Kleinberg, Friedemann Mattern, John C. Mitchell et al. (Hg.): Advances in Knowledge Discovery and Data Mining, Bd. 3918. Berlin, Heidelberg: Springer Berlin Heidelberg (Lecture Notes in Computer Science), S. 450–461.
- Tan, Pang-Ning; Steinbach, Michael; Kumar, Vipin (2005): Introduction to data mining. 1st ed. Boston: Pearson Addison Wesley.
- Tatikonda, Shirish; Parthasarathy, Srinivasan; Kurc, Tahsin (2006): TRIPS and TIDES. In: Philip S. Yu, Vassilis Tsotras, Edward Fox und Bing Liu (Hg.): the 15th ACM international conference on Information and knowledge management. Arlington, Virginia, USA, S. 455.

Termier, A.; Rousset, M.-C.; Sebag, M. (2002): TreeFinder: a first step towards XML data mining. In: 2002 IEEE International Conference on Data Mining. ICDM 2002. Maebashi City, Japan, 9-12 Dec. 2002, S. 450–457.

Terziev, Yordan (2013): Annotating Heterogeneous Document Types. Masterprojekt. Universität Duisburg-Essen, Essen. paluno - The Ruhr Institute For Software Technology.

Ukelson, Jacob P. (2010): What to Do When Modeling Doesn't Work. In: Keith D. Swenson (Hg.): Mastering the unpredictable: Meghan-Kiffer Press, S. 29–39.

van der Aalst, W. M. P.; de Medeiros, A. K. Alves; Weijters, A. J. M. M. (2005): Genetic Process Mining. In: David Hutchison, Takeo Kanade, Josef Kittler, Jon M. Kleinberg, Friedemann Mattern, John C. Mitchell et al. (Hg.): Applications and Theory of Petri Nets 2005, Bd. 3536. Berlin, Heidelberg: Springer Berlin Heidelberg (Lecture Notes in Computer Science), S. 48–69.

van der Aalst, W. M. P.; de Medeiros, A. K. Alves; Weijters, A. J. M. M. (2006): Process Equivalence: Comparing Two Process Models Based on Observed Behavior. In: David Hutchison, Takeo Kanade, Josef Kittler, Jon M. Kleinberg, Friedemann Mattern, John C. Mitchell et al. (Hg.): Business Process Management, Bd. 4102. Berlin, Heidelberg: Springer Berlin Heidelberg (Lecture Notes in Computer Science), S. 129–144.

van der Aalst, W. M. P.; Pesic, M.; Schonenberg, H. (2009): Declarative workflows: Balancing between flexibility and support. In: *Comp. Sci. Res. Dev.* 23 (2), S. 99–113. DOI: 10.1007/s00450-009-0057-9.

van der Aalst, W.M.P. (2009): Using Process Mining to Generate Accurate and Interactive Business Process Maps. In: Witold Abramowicz und Dominik Flejter (Hg.): Business Information Systems Workshops, Bd. 37: Springer Berlin Heidelberg (Lecture Notes in Business Information Processing), S. 1–14. Online verfügbar unter http://dx.doi.org/10.1007/978-3-642-03424-4_1.

van der Aalst, W.M.P. (2011): Process Mining: Discovery, Conformance and Enhancement of Business Processes: Springer Verlag.

van der Aalst, W.M.P.; Weijter, A.J.M.M.; Maruster, L. (2003): Workflow Mining: Discovering process models from event logs. In: *IEEE Transactions on Knowledge and Data Engineering* 16, S. 2004.

van der Aalst, Wil; Adriansyah, Arya; Medeiros, AnaKarlaAlves de; Arcieri, Franco; Baier, Thomas; Blickle, Tobias et al. (2012): Process Mining Manifesto. In: Florian Daniel, Kamel Barkaoui und Schahram Dustdar (Hg.): Business Process Management Workshops, Bd. 99: Springer Berlin Heidelberg (Lecture Notes in Business Information Processing), S. 169–194. Online verfügbar unter http://dx.doi.org/10.1007/978-3-642-28108-2_19.

- van der Aalst, Wil M. P.; Pesic, Maja; Song, Minseok (2010): Beyond Process Mining: From the Past to Present and Future. In: David Hutchison, Takeo Kanade, Josef Kittler, Jon M. Kleinberg, Friedemann Mattern, John C. Mitchell et al. (Hg.): Advanced Information Systems Engineering, Bd. 6051. Berlin, Heidelberg: Springer Berlin Heidelberg (Lecture Notes in Computer Science), S. 38–52.
- van der Werf, J. M. E. M.; van Dongen, B. F.; Hurkens, C. A. J.; Serebrenik, A. (2008): Process Discovery Using Integer Linear Programming. In: PETRI NETS 2008, Bd. 5062. Berlin, Heidelberg: Springer (LNCS), S. 368–387.
- Verbeek, H. M. W.; Buijs, Joos C. A. M.; van Dongen, Boudewijn F.; van der Aalst, Wil M. P. (2011): XES, XESame, and ProM 6. In: Will van der Aalst, John Mylopoulos, Norman M. Sadeh, Michael J. Shaw, Clemens Szyperski, Pnina Soffer und Erik Proper (Hg.): Information Systems Evolution, Bd. 72. Berlin, Heidelberg: Springer Berlin Heidelberg (Lecture Notes in Business Information Processing), S. 60–75.
- Verbeek, H. M. W.; van der Aalst, Wil M. P. (2013): An Experimental Evaluation of Passage-Based Process Discovery. In: Wil van der Aalst, John Mylopoulos, Michael Rosemann, Michael J. Shaw, Clemens Szyperski, Marcello La Rosa und Pnina Soffer (Hg.): Business Process Management Workshops, Bd. 132. Berlin, Heidelberg: Springer Berlin Heidelberg (Lecture Notes in Business Information Processing), S. 205–210.
- Vogel, Oliver (2009): Software-Architektur. Grundlagen - Konzepte - Praxis. 2. Aufl. Heidelberg: Spektrum, Akad. Verl.
- Wang, Chen; Hong, Mingsheng; Pei, Jian; Zhou, Haofeng; Wang, Wei; Shi, Baile (2004): Efficient Pattern-Growth Methods for Frequent Tree Pattern Mining. In: Takeo Kanade, Josef Kittler, Jon M. Kleinberg, Friedemann Mattern, John C. Mitchell, Oscar Nierstrasz et al. (Hg.): Advances in Knowledge Discovery and Data Mining, Bd. 3056. Berlin, Heidelberg: Springer Berlin Heidelberg (Lecture Notes in Computer Science), S. 441–451.
- Wang, Taowei David; Parsia, Bijan (2007): Ontology Performance Profiling and Model Examination: First Steps. In: David Hutchison, Takeo Kanade, Josef Kittler, Jon M. Kleinberg, Friedemann Mattern, John C. Mitchell et al. (Hg.): The Semantic Web, Bd. 4825. Berlin, Heidelberg: Springer Berlin Heidelberg (Lecture Notes in Computer Science), S. 595–608.
- Weidlich, Matthias; Mendling, Jan; Weske, Mathias (2011): A foundational approach for managing process variability. In: Proceedings of the 23rd international conference on Advanced information systems engineering. Berlin, Heidelberg: Springer-Verlag (CAiSE'11), S. 267–282. Online verfügbar unter <http://dl.acm.org/citation.cfm?id=2026716.2026744>.

- Weidt, Stefan (2004): Intraorganisationales Kompetenzmanagement für die Logistikplanung. Dissertation. Universität Dortmund, Dortmund. Fakultät Maschinenbau. Online verfügbar unter <http://dx.doi.org/10.17877/DE290R-15715>.
- Weijters, A. J. M. M.; de Medeiros, A. K. Alves (2006): Process Mining with the HeuristicsMiner Algorithm (BETA Working Paper Series) (WP 166).
- Weijters, A.J.M.M.; Ribeiro, J.T.S.: Flexible Heuristics Miner (FHM). In: SSCI 2011. Paris, France, S. 310–317.
- Wendt, Wolf Rainer (2008a): Case Management im Sozial- und Gesundheitswesen. Eine Einführung. 4., überarb. Aufl. Freiburg im Breisgau: Lambertus. Online verfügbar unter <http://www.socialnet.de/rezensionen/isbn.php?isbn=978-3-7841-1841-3>.
- Wendt, Wolf Rainer (2008b): Case-Management im Sozial- und Gesundheitswesen. Eine Einführung. 4., überarb. Aufl. Freiburg, Br.: Lambertus.
- Wendt, Wolf Rainer (2012): Case Management. In: Christian Thielscher (Hg.): Medizinökonomie. Wiesbaden: Gabler Verlag, S. 505–523.
- Willke, Helmut; Krück, Carsten; Mingers, Susanne (2001): Systemisches Wissensmanagement. Mit 9 Tabellen. 2., Neubearb. Aufl. Stuttgart: Lucius & Lucius (UTB für Wissenschaft: Uni-Taschenbücher, 2047).
- Xinyang Feng; Jianjing Shen; Ying Fan (2009): REST: An alternative to RPC for Web services architecture. In: 2009 First International Conference on Future Information Networks (ICFIN). Beijing, China, S. 7–10.
- Yang, Hedong; van Dongen, B.; ter Hofstede, A.; Wynn, M.; Wang, Jianmin (2012): Estimating completeness of event logs.
- Yang, Qiang (1997): Planning by Decomposition. In: S. Amarel, A. Biermann, L. Bolc, P. Hayes, A. Joshi, D. Lenat et al. (Hg.): Intelligent Planning. Berlin, Heidelberg: Springer Berlin Heidelberg (Artificial Intelligence), S. 85–100.
- Yun Chi; Yirong Yang; Muntz, R. R. (2004): HybridTreeMiner: an efficient algorithm for mining frequent rooted trees and free trees using canonical forms. In: Proceedings. 16th International Conference on Scientific and Statistical Database Management, 2004. Santorini Island, Greece, 21-23 June 2004, S. 11–20.
- Zaki, Mohammed J. (2002): Efficiently mining frequent trees in a forest. In: Osmar R. Zaïane, Randy Goebel, David Hand, Daniel Keim und Raymond Ng (Hg.): the eighth ACM SIGKDD international conference. Edmonton, Alberta, Canada, S. 71.

- Zaki, Mohammed J. (2004): Efficiently Mining Frequent Embedded Unordered Trees. In: *Fundam. Inf.* 66 (1-2), S. 33–52. Online verfügbar unter <http://dl.acm.org/citation.cfm?id=1227174.1227177>.
- Zhang, Li; Bieber, Michael; Song, Min; Oria, Vincent; Millard, David E. (2010): Supplementing virtual documents with just-in-time hypermedia functionality. In: *Int J Digit Libr* 11 (3), S. 155–168. DOI: 10.1007/s00799-011-0065-9.
- Zhao, Peixiang; Yu, Jeffrey (2006): Fast Frequent Free Tree Mining in Graph Databases. In: Sixth IEEE International Conference on Data Mining - Workshops (ICDMW'06). Hong Kong, China, S. 315–319.
- Zhao, Peixiang; Yu, Jeffrey Xu (2007): Mining Closed Frequent Free Trees in Graph Databases. In: Ramamohanarao Kotagiri, P. Radha Krishna, Mukesh Mohania und Ekawit Nantajeewarawat (Hg.): *Advances in Databases: Concepts, Systems and Applications*, Bd. 4443. Berlin, Heidelberg: Springer Berlin Heidelberg (Lecture Notes in Computer Science), S. 91–102.
- Zou, Lei; Lu, Yansheng; Zhang, Huaming; Hu, Rong; Zhou, Chong (2006): Mining Frequent Induced Subtrees by Prefix-Tree-Projected Pattern Growth. In: 2006 Seventh International Conference on Web-Age Information Management Workshops. Hong Kong, China, S. 18.

ABBILDUNGSVERZEICHNIS

Abbildung 1: Grober Ablauf im Process Mining	16
Abbildung 2: Klassifikation von Geschäftsprozessen nach (Loeffeler et al. 1998).....	20
Abbildung 3: Tätigkeitsfelder während der Wissensarbeit nach (North und Güldenber	21
Abbildung 4: Die fünf Phasen des Case Management-Prozesses nach (Wendt 2012).....	30
Abbildung 5: Abstraktes Beispiel eines Case Handling-Modells (Aalst et al. 2005)	35
Abbildung 6: CMMN-Modell eines Schreibprozesses, angelehnt an (Kurz et al. 2015); rechts: Legende.....	36
Abbildung 7: Die drei Typen des Process Mining (van der Aalst et al. 2012).....	41
Abbildung 8: Spaghetti-Prozess nach (van der Aalst 2011).....	43
Abbildung 9: Zyklus des Case-based Reasoning nach (Aamodt und Plaza 1994).....	45
Abbildung 10: Verortung der adCM-Lösungsbausteine im Process Mining-Lifecycle.....	53
Abbildung 11: Lösungsbaustein „adCM-Metamodell“ im Gesamtkontext.....	57
Abbildung 12: Beispiel-Agenda aus dem Rehabilitationsmanagement.....	58
Abbildung 13: Auszug aus dem adCM-Metamodell: Aufbau der Agenda (UML-Klassendiagramm)	58
Abbildung 14: Auszug aus dem adCM-Metamodell: Aufbau des Workspace (UML-Klassendiagramm)	59
Abbildung 15: Auszug aus dem adCM-Metamodell: Annotationen (UML-Klassendiagramm).....	60
Abbildung 16: Auszug aus dem adCM-Metamodell: Templates (UML-Klassendiagramm)	62
Abbildung 17: Definition des adCM-Metamodells und Einsatz der darin enthaltenen Konzepte auf Modell- und Instanzebene.....	65
Abbildung 18: Lösungsbaustein "Monitoring" im Gesamtkontext.....	68
Abbildung 19: Datenmodell für die Ereignis-Protokollierung während der Arbeit mit Artefakten und Agenda	79
Abbildung 20: Facharchitektur für die semantische Verknüpfung.....	82
Abbildung 21: Aufbau der Objekt-Datenstruktur zu einem Fall mithilfe der Ereignisse.....	84
Abbildung 22: adCM Kontext-Hierarchie.....	87
Abbildung 23: Erzeugung des Kontexts	88
Abbildung 24: Definitionen von TF und IDF (Manning et al. 2008)	89
Abbildung 25: Protokollierung von Recherche-Aktivitäten an der zentralen UIA-Suchmaschine.....	90
Abbildung 26: Datenmodell für die Beziehungen zwischen Artefakten.....	91
Abbildung 27: Infrastruktur zur Erstellung und Pflege von Annotationen	92

Abbildung 28: Datenmodell für Annotationen.....	92
Abbildung 29: Zusammenhang zwischen lokaler/globaler Komplexität und Prozessstypen	94
Abbildung 30: Ein Fallbeispiel aus dem Reha-Management.....	105
Abbildung 31: Ähnlichkeit der Fälle 1B-1D zu 1A.....	111
Abbildung 32: Modelle der künstlichen Prozesse, die zur Erzeugung der Ereignisprotokolle L1 (oben) bis L5 (unten) herangezogen wurden	116
Abbildung 33: Prozessmodell zu L ₇ (modelliert als Transitionssystem).....	120
Abbildung 34: Prozessmodell zu L ₈ (mit Heuristics Miner vereinfacht).....	121
Abbildung 35: Prozessmodell zu L ₉ (mit Heuristics Miner vereinfacht).....	122
Abbildung 36: Lösungsbaustein „Discovery“ im Gesamtkontext.....	126
Abbildung 37: Ob diese beiden Bäume gleich sind oder nicht, hängt davon ab, ob sie geordnet sind	131
Abbildung 38: Beispielbaum mit Teilbäumen in Anlehnung an (Jiang et al. 2013)	132
Abbildung 39: Mehrschichtiges Datenmodell für das Discovery	138
Abbildung 40: Die wesentlichen Verarbeitungsschritte des adCM Miners.....	146
Abbildung 41: Das "Wachsen" von Teilbäumen als grundsätzliches Prinzip von FSM-Algorithmen.....	149
Abbildung 42: Datenfluss des Experiments, beginnend mit dem Ereignisprotokoll, bis zum Vergleich der Qualitätsmetriken (oben: adCM-Teststrecke; unten: Process Mining-Teststrecke)	154
Abbildung 43: Schritte zur Vorbereitung und Durchführung der Evaluation	154
Abbildung 44: Unpräzises Modell für die Beschreibung eines Ereignisprotokolls	155
Abbildung 45: Präzises Modell mit schlechter Fitness bezogen auf das Protokoll aus Abbildung 44.....	156
Abbildung 46: Venn-Diagramm zur Veranschaulichung der Zusammenhänge zwischen Fitness und Precision im Kontext von adCM.....	157
Abbildung 47: Modell mit schlechter (links) und guter (rechts) Abstraktion zur Beschreibung des Ereignisprotokolls aus Abbildung 44.....	158
Abbildung 48: Ein kompliziertes (links) und ein einfaches (rechts) Modell für denselben Prozess.....	159
Abbildung 49: Anwendung der Kreuzvalidierung im Experimentaufbau	161
Abbildung 50: Transformation der Traces im Ereignisprotokoll in Agendastrukturen	164
Abbildung 51: Auswirkungen von Schwellwertänderungen auf die Anzahl erzeugter Templates.....	168
Abbildung 52: Beispiel für das "Verschlucken" von kleinen Teilbäumen durch größere, seltenere Bäume.....	169
Abbildung 53: Die Unterstützungsfunktionen im adCM-Gesamtkontext.....	176
Abbildung 54: Integrationskonzept der adCM-Lösungsbausteine in eine Unternehmensarchitektur (UML-Komponentendiagramm).....	182
Abbildung 55: Kopplung von Client- und Service Framework mit einer ereignisgesteuerten Architektur (UML-Komponentendiagramm).....	185

Abbildung 56: Architektur des Monitoring-Services (UML Komponentendiagramm).....	187
Abbildung 57: UML Kommunikationsdiagramm für den Aufruf des AgendaMiners (UML Komponentendiagramm).....	190
Abbildung 58: Darstellung der Interaktion mithilfe eines UML-Kommunikationsdiagramms.....	192
Abbildung 59: Vereinfachte Darstellung des Lebenszyklus des dreistufigen Recommender-Algorithmus (UML Zustandsdiagramm).....	195
Abbildung 60: Architektursicht auf den einheitlichen Datenzugriff (UML Komponentendiagramm).....	197
Abbildung 61: Adapter-Architektur zur Anbindung verschiedener Synchronisationsmechanismen.....	201
Abbildung 62: Binäre (links) und n-äre (rechts) Beziehungen zwischen Artefakten auf dem Workspace.....	202
Abbildung 63: Integrierte Darstellung mehrerer Workspaces mithilfe von Auswahlboxen (Flock 2014).....	204
Abbildung 64: Integrierte Darstellung mehrerer Workspaces entlang ihrer hierarchischen Beziehung (Flock 2014).....	205
Abbildung 65: Zuordnung der Artefakte eines integrierten Workspace durch Hervorhebung (Flock 2014).....	206
Abbildung 66: Integrierte Darstellung mehrerer Workspaces mithilfe einer Treemap (Flock 2014).....	207
Abbildung 67: Architektur des Annotations-Service.....	209
Abbildung 68: Integration der Annotationsfunktion im Browser (hier: Firefox).....	210
Abbildung 69: Integration der Annotationsfunktion in Microsoft Word.....	211
Abbildung 70: Architektur des ACTIVE Knowledge Workspace (UML-Komponentendiagramm).....	215
Abbildung 71: Komponenten eines PAIS nach (Reichert und Weber 2012, S. 32) als UML-Komponentendiagramm.....	217
Abbildung 72: Integration von Unterstützungssystemen im Process Mining-Umfeld nach (van der Aalst et al. 2010).....	218
Abbildung 73: Komponenten der DYONIPOS Architektur nach (Rath 2007) als UML Komponentendiagramm.....	220
Abbildung 74: Verlauf des Fallbeispiels entlang der Zeitachse mit unvorhergesehenem Ereignis (Blitz).....	225
Abbildung 75: Gliederung der Agenda, die auf 60% der Fälle zutrifft.....	226
Abbildung 76: Wiederkehrendes Muster bei der Behandlung posttraumatischer Belastungsstörungen.....	226
Abbildung 77: Zustand des Workspace nach dem Start des neuen Falls.....	228
Abbildung 78: Strukturierung der Rechercheergebnisse zu den Therapiemaßnahmen.....	230
Abbildung 79: Reflektieren und Recherchieren von Therapiemaßnahmen mit einem adCM-Werkzeug.....	231
Abbildung 80: Abgleich zwischen der aktuellen Agenda und dem Template.....	232
Abbildung 81: Dialog des Recommender-Algorithmus, um das Template vorzuschlagen.....	233
Abbildung 82: Finale Agenda des Fallbeispiels mit Angaben zur Herkunft und Verwendung.....	234

TABELLENVERZEICHNIS

Tabelle 1: Einsatzgebiete des Case Managements nach (Swenson 2010b; Wendt 2008b; Mühlum und Gödecker-Geenen 2003; McCauley 2010).....	26
Tabelle 2: Beispiel einer Protokoll-Datei (Prozess "Lehrveranstaltung planen und durchführen").....	80
Tabelle 3: Zwei Extrembeispiele zur Veranschaulichung von TV.....	98
Tabelle 4: Zwei Beispiel-Ereignisprotokolle zur Erläuterung der Trace-Disparität	101
Tabelle 5: Vergleich der Rand-Dispersität zweier unterschiedlicher Ereignisprotokolle (n=15)	104
Tabelle 6: Abdeckung der Ereignisse durch den Testfall.....	107
Tabelle 7: Überprüfung, ob die Ereignisse vollständig protokolliert wurden.....	108
Tabelle 8: Ergebnis des Abgleichs zwischen dem dateibasierten und dem semantischen Ereignisprotokoll.....	110
Tabelle 9: Matrix mit den Kosinus-Ähnlichkeiten zwischen den Fällen 1A-1D	112
Tabelle 10: Matrix mit den Kosinus-Ähnlichkeiten zwischen den Fällen 2A-2D	114
Tabelle 11: Details zu den synthetischen Beispielprotokollen	115
Tabelle 12: Auswertung der Komplexitätsmetriken für die künstlichen Ereignisprotokolle L ₁ -L ₆	117
Tabelle 13: Details zu den Beispielprotokollen aus den beiden niederländischen Krankenhäusern	120
Tabelle 14: Auswertung der Komplexitätsmetriken für die künstlichen Ereignisprotokolle L ₇ , L ₈ und L ₉	122
Tabelle 15: Übersicht über Frequent Tree Mining-Algorithmen	133
Tabelle 16: Gegenüberstellung der Qualitätsmetriken auf der Grundlage von insgesamt 1143 Fällen	170
Tabelle 17: Szenarien zur Konkretisierung architektureller Anforderungen nach (Bass et al. 2013)	178
Tabelle 18: Szenario zur Konkretisierung architektureller Anforderungen an den Monitoring-Service.....	184
Tabelle 19: Szenario zur Konkretisierung architektureller Anforderungen an den Discovery-Service	188

EIDESSTATTLICHE ERKLÄRUNG

Ich gebe folgende eidesstattliche Erklärung ab:

Ich erkläre hiermit, dass ich die vorliegende Arbeit selbständig ohne unzulässige Hilfe Dritter verfasst, keine anderen als die angegebenen Quellen und Hilfsmittel benutzt und die Arbeit keiner anderen Stelle zu Prüfungszwecken vorgelegt habe.

Marian Benner-Wickner

Dortmund, den 16. Dezember 2016